Tartalomjegyzék

1. Bevezetõ

Az algebra és a számelmélet alkalmazásának manapság igen közkedvelt ága a kriptográfia, a titko: tudománya, amely már a római idõktõl kezdve ismeretes. Állítólag Julius Caesar hadjáratai során kó(üzeneteket küldött hadvezéreinek. (Nem véletlenül volt olyan sikeres.) Ám a kutatások központjába mégiscsak a 20. század végén került. A technika rohamléptû fejlõdésével lehetõség nyílik arra, hogy emberek az internet és a mobil kommunikáció eszközeivel kommunikáljanak vagy információt cseréljenek. Azonban a kibontakozófélben lévõ információs társadalomban maga az információ igen fontos érték, amelynek a biztonságos kezelésérõl és cseréjérõl gondoskodni kell. Nem véletlen tehát, hogy a kriprográfia tudománya, alkalmazása egyre inkább teret hódít. Vége már azoknak az idõknek amikor a titkosítás tudományát csak és kizárólag a hadiipar használta fel. Manapság egyre inkább ke: mindennapos élet részévé válni, az élet minden területén, gondoljunk akár a banki kereskedelemre, a számítástechnikára, a magánkommunikációra, vagy akár a mindennapos „bolti" kereskedelemre.

Hazánkban is egyre inkább kezd teret hódítani a elektronikus kereskedelem, ahol a szobában egy számítógép mellett ülve tudunk vásárolni, eladni, üzletet kötni. Hogyan is néz ki egy ilyen tranzakció Mondjuk az interneten barangolva meglátunk egy érdekesnek ígérkezõ könyvet. Elolvassuk az ajánlá róla, megtetszik. Most várjuk meg, amíg megjelenik a boltok polcain, vagy ha lehetõség van rá rende meg azonnal? A kényelmi szempontok azt diktálják, hogy még a székünkbõl se álljunk fel és azonnal rendeljük meg a könyvet. Kitöltve és elküldve egy rendelési formanyomtatványt, amely tartalmazza a bankkártyánk, bankszámlaszámunk számát, máris hozzájutottunk a frissen megtetszett könyvhöz. Az élet viszont nem ilyen szép. A hálón „keringõ" formanyomtatványhoz gyakorlatilag bárki hozzáférhe onnan az adatainkat eltulajdonítva igen kellemetlen meglepetést okozhat nekünk akkor, amikor a bankkártyánkon lévõ pénz felõl érdeklõdünk. Tehát tiszta sor, hogy megpróbáljuk adatainkat kódolt formában elküldeni.

Egy másik szemszögbõl nézve a fent említett dolgokat: egy titkosított üzenet tartalma lehet törvénysértõ vagy esetleg államellenes. Ekkor, ha megfelelõen erõs kódolást alkalmaztunk, akkor esélyük sincs az illetékes szerveknek ezen üzenetek tartalmához hozzájutni. Tehát két dolog van ami verseng egymással: az alapvetõ emberi jogok a törvényi szabályozás ellenõrizhetõségével. Országokt függõen több megoldási mód létezik: például Franciaországban büntetik azt is, ha valaki a megengedettnél erõsebb (40 bites kulcsnál nagyobb) titkosítási módszert alkalmaz. Máshol olyan törvényt készülnek hozni, hogy a kódolásnál alkalmazott kulcsokat be kell nyújtani a megfelelõ állam szervekhez.

Dolgozatomban csak pártatlanul, az elõzõ kérdésekben nem állásfoglalva, szeretnék bemutatni bizonyos a kriptográfiai tudomány hátterében álló matematikai eszközt és megmutatni a titkosírás tudományának néhány módszerét. Így a második fejezetben néhány alapvetõ számelméleti és algebra tényt említek meg, amelyek a titkosírás tudományának alapkövei. Ezt követõen néhány hagyományo kódolási módszert mutatok be példákkal illusztrálva, majd az amerikai Nemzeti Szabványügyi Hivata (NBS) által szabványnak minõsített Data Encryption Standardet. Itt jegyezném meg, hogy a dolgozatomban említett példák didaktikai jellegûek, a módszerek szemléltetését, jobb megértését és a biztonságos kódolást szolgálják. A negyedik fejezetben egy egészen forradalmi újítást írok le: a nyilvános kulcsú kriptorendszerek ötletét, illetve ezen alapötlet implementálásának néhány módját: a: RSA-t és az ElGamal sémát. Ezt követõen egy új területre kalauzolom az olvasót: az elliptikus görbe területére, az elliptikus görbék pontjai felett definiálható csoport elméletébe. Végezetül az elliptikus görbéken alapuló kódolási rendszereket ismerhetjük meg, összehasonlítva biztonságukat más, nem elliptikus görbéket alkalmazó kriptorendszerekével. A függelékben pedig megtalálhatók algoritmuso ábrák, táblázatok, amelyek a gyakorlati alkalmazásokhoz nyújtanak segítséget.

Végezetül még egy gondolatot: a kriptográfia egy ma is igen dinamikusan fejlõdõ tudományág. Jelenlegi kutatások középpontjában a meglévõ rendszerek biztonságával kapcsolatos feltevések (pélc az RSA feltörése ekvivalens-e a kódolásnál használt modulus faktorizációjával, vagy sem) igazolása vagy cáfolása, illetve olyan matematikai struktúrák keresése, elméleteik felállítása, amelyek újabb kódolási módszerek alapjául szolgálhatnak (például az elliptikus görbék elmélete.) Ennek a komplex tudományágnak elsõsorban matematikai perspektívából való megközelítését szolgálja dolgozatom.

2. Számelméleti és algebrai bevezetõ

Legelõször is tekintsük át azokat az algebrai és számelméleti fogalmakat, tételeket és eszközöket, amelyek segítségünkre lesznek a kriptográfiai kérdések egzakt, matematikai nyelven való megfogalmazásához, illetve ezen kérdések megválaszolásához.

Igen hosszadalmas lenne teljesen az alapoktól kezdve felépíteni a számelméleti és az algebrai hátte így feltételezzük, hogy a számelmélet alapfogalmai és alaptételei; mint például egység, felbonthatatla prím, oszthatóság, maradékos osztás tétele stb. ismertek.

2.1. A számelmélet néhány fogalma, eredménye

Def: Legyen $m > 1$ egész és $a,b \in$ Z. Ekkor $a \equiv b$ mod m (a kongruens b modulo m), ha $m| a$ - b

Megjegyzés: Legyen $a,b \in$Z és a maradékos osztás tételét alkalmazva írjuk fel a-t és b-t a követke alakban:

$$a = q_1m + r_1 \text{ és } b = q_2m + r_2, \text{ ahol } 0 \leq r_1, r_2 \leq m-1.$$

Azaz, $a \equiv b$ mod m, akkor és csak akkor, ha $r_1 = r_2$. A továbbiakban jelölje a mod m az a-nak m-r vett osztási maradékát; jelen esetben r_1-et (azaz a-t redukáltuk modulo m).

Def: Legyen $m > 1$ rögzített egész és $r \in$ Z. Tekintsük az $mt + r$ alakú egészeket, ahol $m,t \in$ Z. E egy modulo m maradékosztály.

Megjegyzés: Modulo m maradékosztály redukált maradékosztály, ha minden eleme relatív prím m-h Azaz, ha lnko(m, r) = 1, akkor az r mod m maradékosztály redukált.

Def: Legyen $m > 1$ rögzített egész. $\mathbf{Z}_m$ = {a mod m maradékosztályok, ahol $a \in$ Z.}

Ezek után nézzük meg a mûveleteket modulo m $\mathbf{Z}_m$-ben:

Az összeadás és a szorzás teljesen ugyan úgy megy, mint ahogyan az egészek körében megszoktu azzal a különbséggel, hogy az eredményt redukáljuk modulo m. A következõ tulajdonságai vannak a modulo m összeadásnak és szorzásnak:

1. Minden $a,b \in \mathbf{Z}_m$-re $a + b \in \mathbf{Z}_m$, azaz összeadásra zárt

2. Minden $a,b \in \mathbf{Z}_m$-re $a + b = b + a$, azaz az összeadás kommutatív

3. Minden $a,b,c \in \mathbf{Z}_m$-re $(a + b) + c = a + (b + c)$, azaz az összeadás asszociatív

4. Minden $a \in \mathbf{Z}_m$-re $a + 0 = 0 + a = a$, azaz 0 az additív egységelem

5. Minden $a \in \mathbf{Z}_m$-re egyértelmûen létezik $b \in \mathbf{Z}_m$ (és $b = m - a$), hogy $a + b = b + a = 0$, azaz létezik additív inverz

6. Minden $a,b \in \mathbf{Z}_m$-re $ab \in \mathbf{Z}_m$, azaz szorzásra zárt

7. Minden $a,b \in \mathbf{Z}_m$-re $ab = ba$, azaz a szorzás kommutatív

8. Minden $a,b,c \in \mathbf{Z}_m$-re $(ab)c = a(bc)$, azaz a szorzás asszociatív

9. Minden $a \in \mathbf{Z}_m$-re $a1 = 1a = a$, azaz 1 a multiplikatív egységelem

10. Minden $a,b,c \in \mathbf{Z}_m$-re $(a + b)c = ac + bc$, és $a(b + c) = ab + ac$, azaz a szorzás disztributív az összeadásra nézve.

Megjegyzés: Az absztrakt algebra nyelvén megfogalmazva $\mathbf{Z}_m$ csoport az összeadásra nézve, az 1. é 3-5. tulajdonságok miatt. Ha teljesül a 2. tulajdonság is egy csoportra, akkor azt mondjuk, hogy a csoport Abel-csoport, azaz kommutatív.

Def: Azt az algebrai struktúrát, amely rendelkezik a fenti tulajdonságok mindegyikével, kommutatív (egységelemes) gyûrûnek nevezzük.

Megjegyzés: Mivel minden $a \in \mathbf{Z}_m$-nek létezik additív inverze, ezért $\mathbf{Z}_m$ elemei között kivonást is végezhetünk. Tehát minden $a,b \in \mathbf{Z}_m$ -re: $a - b = a + m - b$, ami ugyan az, mintha kiszámolnánk a - értékét és redukálnánk modulo m.

Def: Egy $a \in \mathbf{Z}_m$ elem multiplikatív inverze az $a^{-1} \in \mathbf{Z}_m$ elem, amelyre: $aa^{-1} \equiv a^{-1}a \equiv 1 \mod m$.

Észrevétel: A fenti feltételek nem szólnak semmit a multiplikatív inverz létezésérõl. Általában nem is igaz, hogy minden $a \in \mathbf{Z}_m$-nek létezik multiplikatív inverze. Nézzük meg, hogy mi a feltétele a multiplikatív inverz létezésének: Meg kell oldanunk a következõ egy ismeretlenes, lineáris kongruenciát: $ax \equiv 1 \mod m$, ahol $a,m \in \mathbf{Z}_m$ adottak és keressük $x \in \mathbf{Z}_m$-t.

Mi a feltétele általában az $ax \equiv b \mod m$ lineáris kongruencia megoldhatóságának?

A $ax \equiv b \mod m$ kongruencia megoldható akkor és csak akkor, ha $m \mid ax - b$-t, azaz $\exists y \in \mathbf{Z}$ úgy hogy $my = ax - b$, azaz létezik megoldása az $ax - my = b$ két ismeretlenes, lineáris diofantikus egyenletnek. Ennek pedig akkor és csak akkor létezik megoldása, ha lnko$(a, m) \mid b$.

Azaz esetünkben az $ax \equiv 1 \mod m$ kongruenciának létezik megoldása, ha lnko$(a, m) \mid 1$, azaz a és relatív prímek.

Következmény: a-nak létezik multiplikatív inverze modulo m akkor és csak akkor, ha a és m relatív prímek.

Megjegyzés: Abban az esetben, ha p prím, akkor minden $0 \neq a \in \mathbf{Z}_p$ relatív prím p-hez, azaz minden $a \in \mathbf{Z}_p$-nek létezik multiplikatív inverze, azaz $\mathbf{Z}_p$ p prím esetén testet alkot az összeadás és szorzás mûveletével.

Hogyan állíthatjuk elõ $a \in \mathbf{Z}_m$ multiplikatív inverzét, ha a, m relatív prímek? Létezik egy kiterjesztett euklideszi algoritmus néven ismert eljárás, amely segítségével meghatározhatjuk egy $a \in \mathbf{Z}_m$ elem multiplikatív inverzét, ha létezik. Elõször nézzük meg az eredeti euklideszi algoritmust, ami két szám: r_0, r_1 legnagyobb közös osztójának meghatározására szolgál:

Legyen $0 < r_0, r_1 \in \mathbf{Z}$. Maradékos osztás tétele szerint létezik $q_i, r_i \in \mathbf{Z}$ úgy, hogy:

$$r_0 = q_1 r_1 + r_2, \text{ ahol } 0 < r_2 < r_1$$
$$r_1 = q_2 r_2 + r_3, \text{ ahol } 0 < r_3 < r_2$$
$$\vdots$$
$$r_{k-2} = q_{k-1} r_{k-1} + r_k, \text{ ahol } 0 < r_k < r_{k-1}$$
$$r_{k-1} = q_k r_k.$$

Észrevétel: Az eljárás véges sok lépésben befejezõdik, hiszen: $r_1 > r_2 > \cdots > r_k$.
Nem nehéz bizonyítani, hogy: $\mathrm{lnko}(r_0, r_1) = \mathrm{lnko}(r_1, r_2) = \cdots = \mathrm{lnko}(r_{k-1}, r_k) = r_k$.

Amellett, hogy az euklideszi algoritmust két szám legnagyobb közös osztójának meghatározására használjuk, közvetetten használható egy $a \in \mathbf{Z}_m$ elem multiplikatív inverzének meghatározására is. Definiáljuk a következõ rekurzív sorozatot:

$$t_0 = 0$$
$$t_1 = 1$$
$$t_j = t_{j-2} - q_{j-1} t_{j-1} \bmod a,\ j \geq 2\text{-re},$$

ahol q_j-k az euklideszi algoritmusban definiált értékek.

Tétel: $0 \leq j \leq k$ -ra kapjuk, hogy $r_j \equiv t_j r_1 \bmod r_0$, ahol q_j és r_j értékek az euklideszi algoritmusban definiált értékeknek illetve a t_j-k a fent definiált rekurzív sorozat értékeinek felelnek meg.

biz: Teljes indukció j-re. A feltétel triviális $j = 0$ és $j = 1$ -re. Tegyük fel, hogy a feltétel igaz $j = i$ - re és $j = i - 2$ -re. Megmutatjuk, hogy igaz $j = i$ -re is. Az indukciós feltétel szerint: $r_{i-2} \equiv t_{i-2} r_1 \bmod r_0$ és $r_{i-1} \equiv t_{i-1} r_1 \bmod r_0$. Így ki tudjuk számolni r_i-t: $r_i = r_{i-2} - q_{i-1} r_{i-1} \equiv t_{i-2} r_1 - q_{i-1} t_{i-1} r_1 \equiv (t_{i-2} - q_{i-1} t_{i-1}) r_1 \equiv t_i r_1 \bmod r_0$. Ezzel beláttuk az indukciós feltevést, azaz igazoltuk az állítást.

Következmény: Tegyük fel, hogy $\mathrm{lnko}(r_0, r_1) = 1$. Ekkor $t_k = r_1^{-1} \bmod r_0$.

A 7.1. Függelékben megtalálható a kiterjesztett euklideszi algoritmus leírása.

Def: Legyen $2 \leq m \in \mathbf{Z}$. Ekkor az m-hez relatív prímek számát jelölje: $\phi(m)$.

Észrevétel: Ekkor a mod m redukált maradékosztályok száma: $\phi(m)$.

Def: Legyen $m > 1$ rögzített egész. Ekkor $a_1, a_2, \dots, a_r$ redukált maradék rendszer modulo m, ha minden redukált maradékosztályhoz egyértelmûen létezik a_i, amely eleme annak a redukált maradék osztálynak és $r = \phi(m)$.

Tétel: Legyen $1 \le n \in \mathbf{Z}$ kanonikus alakja: $n = \prod_{i=1}^{k} p_i^{r_i}$, ahol p_i-k különbözõ prímek és $r_i > 0$, $1 \le i \le k$ -ra. Ekkor $\phi(n) = \prod_{i=1}^{k} (p_i^{r_i} - p_i^{r_i - 1})$.

Euler-Fermat tétel: Legyen $1 \le m \in \mathbf{Z}$ és $c \in \mathbf{Z}$ úgy, hogy lnko(c, m) = 1. Ekkor $c^{\phi(m)} \equiv 1$ mod n

Megjegyzés: speciálisan m=p prím esetén kapjuk: $c^{p-1} \equiv 1$ mod p, ami a **Kis-Fermat tétel**.

Def: Legyen $p > 2$ prím, és $1 \le x \le p-1$ egész. Azt mondjuk, hogy x kvadratikus maradék modulo ha az $y^2 \equiv x$ mod p kongruenciának létezik $y \in \mathbf{Z}_p$ megoldása. Hasonlóan, ha az elõbbi kongruenciá nincs megoldása, akkor azt mondjuk, hogy x kvadratikus nem maradék.

Euler-lemma: Adott $p > 2$ prím. Ekkor $x \ne 0$ kvadratikus maradék mod p akkor és csak akkor, ha $x^{(p-1)/2} \equiv 1$ mod p.

biz: Elõször tegyük fel, hogy az $x \equiv y^2$ mod p kongruenciának létezik megoldása mod p. Ekkor: $x^{(p-1)/2} \equiv (y^2)^{(p-1)/2} \equiv y^{p-1} \equiv 1$ mod p. Az elsõ kongruencia a feltétel miatt, a második a **Kis Fermat tétel** miatt igaz.

Másodszor tegyük fel, hogy $x^{(p-1)/2} \equiv 1$ mod p. Legyen b primitív elem modulo p. Ekkor létezik $i \ge 1$ egész, úgy hogy $x \equiv b^i$ mod p. Ekkor: $x^{(p-1)/2} \equiv (y^2)^{(p-1)/2} \equiv y^{p-1} \equiv 1$ mod p. Mivel i rendje (p - 1), ezért (p - 1) osztja $i(p$ - 1)/2-t. Ezért i páros, és x-nek létezik négyet gyöke modulo p; nevezetesen $\pm b^{i/2}$.

Def: Legyen $p > 2$ prím. Minden $a \ge 0$ egészre definiáljuk a $\left(\frac{a}{p}\right)$ Legrende-szimbólumot a következõképpen:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, \text{ ha a } \equiv 0 \text{ mod p} \\ 1, \text{ ha a kvadratikus maradék mod p} \\ -1, \text{ ha a kvadratikus nem maradék mod p} \end{cases}$$

Következmény: Az Euler-lemma egy egyszerû következménye:

Ha $p > 2$ prím, egész, akkor: $\left(\frac{a}{p}\right) \equiv a^{(p-1)/2}$ mod p.

Def: Legyen $n > 2$ páratlan egész és legyen n prímtényezõs felbontása: $n = p_1 p_2 \cdots p_r$. Minden $b \geq$ egészre definiáljuk a $\left(\frac{b}{n}\right)_J$ Jacobi-szimbólumot a következõképpen: $\left(\frac{b}{n}\right)_J = \prod_{i=1}^{r}\left(\frac{b}{p_i}\right)$.

Tétel: Legyen $m, n > 2$ páratlan egész és $a, b \geq 0$ egészek. Ekkor a következõ tulajdonságok igaza

1. Ha $a \equiv b$ mod n, akkor $\left(\frac{a}{n}\right)_J = \left(\frac{b}{n}\right)_J$;
2. $\left(\frac{ab}{n}\right)_J = \left(\frac{a}{n}\right)_J \left(\frac{b}{n}\right)_J$;
3. $\left(\frac{-1}{n}\right)_J = \begin{cases} 1, \text{ ha n} \equiv 1 \bmod 4 \\ -1, \text{ ha n} \equiv -1 \bmod 4 \end{cases}$;
4. $\left(\frac{2}{n}\right)_J = \begin{cases} 1, \text{ ha n} \equiv \pm 1 \bmod 8 \\ -1, \text{ ha n} \equiv \pm 3 \bmod 8 \end{cases}$;
5. $\left(\frac{m}{n}\right)_J = \begin{cases} -\left(\frac{n}{m}\right)_J, \text{ ha m} \equiv \text{n} \equiv 3 \bmod 4 \\ \left(\frac{n}{m}\right)_J, \text{ különben.} \end{cases}$

Ezen tétel alkalmazásával, az euklideszi algoritmus mintájára, meg tudjuk határozni bármely Jacob szimbólum értékét. A tétel bizonyítása technikai jellegû, ezért azt mellõzném.

Végezetül az elemi prímszámelmélet egy tétele:

Nagy prímszámtétel: Legyen $\pi(x)$ az x-nél nem nagyobb prímszámok száma. Ekkor: $\lim_{x\to\infty} \frac{\pi(x)}{x/\log x} = 1$, azaz $\pi(x)$ és $x/\log x$ asszimptotikusan egyenlõek.

Ezek után a hasznos jobbára számelméleti tételek után nézzük meg az absztakt algebra néhány eredményét.

2.2. Az absztrakt algebra néhány fogalma, eredménye

Def: Legyen G (véges) csoport és legyen 1_G a csoport egységeleme. Ekkor $g \in G$ -nek a rendje az legkisebb pozitív n egész, amelyre
$g^n = 1_G$.

Def: Legyen G véges csoport. Ekkor G csoport rendje megegyezik a csoport elemszámával.

Lagrange tétel: Legyen G véges és legyen $g \in G$. Ekkor g rendje osztója a csoport rendjének.

Megjegyzés: Speciálisan, ha *G* mod *m* redukált maradékosztályok multiplikatív csoportja, akkor a **Lagrange tételt** alkalmazva *G*-re az **Euler-Fermat tételt** kapjuk.

Def: Egy csoport ciklikus, ha egyetlen elem (összes, egész kitevõs) hatványaiból áll. Ezt az elemet a csoport generáló elemének nevezzük.

Def: Legyen *G* véges, ciklikus csoport. Ekkor a csoport (bármely) generátorelemét a csoport primit elemének nevezzük.

Tétel: Véges test nem nulla elemei a szorzásra nézve ciklikus csoportot alkotnak.

Következmény: Ha p prím, akkor a $\mathbf{Z}_p^*$ ciklikus.

Legyen p prím és legyen a primitív elem modolo p. Ekkor minden $b \in \mathbf{Z}_p^*$ elem egyértelmûen elõáll a^i alakban, ahol $0 \le i \le p-2$. Ekkor $b = a^i$ rendje: $\frac{p-1}{\mathrm{lnko}(p-1, i)}$. Tehát, b akkor és csak akkor primitív elem modulo p, ha $\mathrm{lnko}(p-1, i) = 1$. Ebbõl következik, hogy modulo p primitív elemek szám $(p - 1)$.

2.3. Az ismételt négyzetre emelés módszere

Végezetül ismerkedjünk meg az ismételt négyzetre emelés módszerével, amellyel hatékonyan tudunl hatványozni mod n.

Feladat a következõ lenne: meg kellene határozni a 9726^{3533} mod 11413 értéket. Egy olyan eljárá keresünk amely a lehetõ legkevesebb lépésben, a lehetõ legrövidebb idõ alatt állítja elõ a megfelelõ értéket. A módszer alapgondolata (hatványozás helyett szorzás) már az egyiptomi Rhind-papiruszon megtalálható; tehát kb. 3000 éves receptrõl van szó.

Elõször határozzuk meg 3533 kettes számrendszerbeli értékét: $3533 = 110111001101_2$. Ezt köve elkezdjük meghatározni az alap négyzeteinek értékét, persze mod 11413. Azaz:

$$3533 \equiv 3533 \bmod 11413$$
$$3533^2 \equiv 7680 \bmod 11413$$
$$3533^4 \equiv 7680^2 \equiv 16 \bmod 11413$$
$$\vdots$$
$$3533^{2048} \equiv 2954 \bmod 11413$$

Ezt követõen a megfelelõ hatványokat összeszorozva és redukálva 11413 megkapjuk a helyes eredményt, azaz 9726^{3533} mod $11413 = 5761$.

Az ismételt négyzetre emelés algoritmusa megtalálható a 7.2. Függelékben.

3. Titkos kulcsú kriptorendszerek

3.1. Bevezetés a kriptográfiába

Mit is jelent az a szó kriptográfia? Az üzenetek valamilyen módon való kódolásának, titkosításána tudománya. Ez alapvetõen a következõt jelenti: adott két ember, nemzetközi irodalomban szokás õk Alice és Bob néven emlegetni, akik egy olyan közegben szeretnének kommunikálni, amely nem biztonságos, azaz az üzeneteiket bárki elolvashatja. (Ez a csatorna lehet akár telefon, internet vagy bármely más kommunikációs közeg.) Õk mégis olyan módon szeretnének információt cserélni, hogy illetéktelenek ne értsék meg az üzeneteket. A megoldás roppant egyszerû:

Alice nem az eredeti üzenetet küldi el Bobnak, hanem egy elõre meghatározott kulcs segítségével kódolt üzenetet küld. Ha bárki, aki látja a kódolt üzenetet, de nem ismeri a kulcsot, amellyel kódoltá nem fogja megérteni az üzenet tartalmát. Bob viszont, aki ismeri a kulcsot egyszerûen dekódolja az üzenetet.

Ennek az elgondolásnak a formális leírásához használjuk a következõ definíciót:

Def: A kriptorendszert a következõ ötös írja le $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, ahol a következõk teljesülnek:

1. $\mathcal{P}$ a lehetséges üzenetek véges halmaza.
2. $\mathcal{C}$ a lehetséges kódolt üzenetek véges halmaza.
3. $\mathcal{K}$ a kulcskészlet, lehetséges kulcsok véges halmaza.
4. Minden $K \in \mathcal{K}$-ra létezik egy $e_K \in \mathcal{E}$ kódolási szabály, és egy hozzá tartozó $d_K \in \mathcal{D}$ dekódolási szabály. Az $e_K \in \mathcal{P} \to \mathcal{C}$ és $d_K \in \mathcal{C} \to \mathcal{P}$ függvényekre teljesül, hogy $\forall x \in \mathcal{P}$-re: $d_K(e_K(x)) = x$.

Nézzük meg, hogy miként is néz ki a fent ismertetett eljárás a definicióban használt terminológiák használatával:

Elõször Alice és Bob közösen, véletlenszerûen választanak egy kulcsot: $K \in \mathcal{K}$-t. Ezt úgy teszik meg, hogy csak õk ketten ismerjék a választott kulcsot. Ezek után, tegyük fel, hogy Alice üzenni aka Bobnak. Az üzenetet egyértelmûen azonosítható egységekre, karakterekre tördeljük. Tegyük fel, ho az üzenet a következõ karakterekbõl áll: $x = x_1 \ldots x_n$, ahol $n \geq 1$ és $x_i \in \mathcal{P}$ teljesül $1 \leq i \leq n$ -re.

Az üzenet x_i karakterei a választott kulcs által meghatározott kódolási szabály, e_K, szerint kerülnek kódolásra. Így Alice meghatározza az $y_i = e_K(x_i)$ értékeket $1 \leq i \leq n$ -re és ezzel elõállít titkosított szöveget: $y = y_1 \ldots y_n$, amit elküld Bobnak.

Miután Bob megkapja az $y = y_1 \ldots y_n$ szöveget, a d_K dekódoló függvénnyel meg tudja határozr az $x = x_1 \ldots x_n$ értékeket, amely az eredeti üzenet.

Megjegyzés: Nyilvánvaló, hogy e_K injektív függvény, különben nem lesz egyértelmûen dekódolhat üzenet. Például, ha $y = e_K(x_1) = e_K(x_2)$ teljesül, ahol $x_1 \neq x_2$, akkor Bob nem tudja eldönteni, hc az eredeti üzenetben x_1 vagy x_2 volt-e. Vegyük észre azt is, hogy ha $\mathcal{P} = \mathcal{C}$, akkor e_K a $\mathcal{P}$ egy permutációja.

3.2. Titkos kulcsú kriptorendszerek

Mit jelent az a jelzõ, hogy titkos kulcsú egy kriptorendszer? A fent említett gondolatmenet legfontosabb része, hogy Alice és Bob által közösen választott kulcsot titkosan kell kiválasztani és diszkréten kell kezelni. Ugyanis a kulcs birtokában a kódolt üzenetek pillanatok alatt megfejthetõk.

Most nézzünk meg egy-két igen alapvetõ titkos kulcsú kriptorendszert:

3.2.1. Eltolásos módszer

Már Julius Ceasar idejében ismert volt ez a módszer (állítólag õ volt az, aki elõször alkalmazta): Például az angol abc minden egyes betûje helyett, tegyük fel, hogy a rákövetkezõ harmadikat haszná Így aki nem ismeri a cselt, az nem tudja majd elolvasni üzenetünket.

Formálisan leírva a rendszert:

> Legyen $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbf{Z}_{26}$. Minden $x, y \in \mathbf{Z}_{26}$ -ra és $0 \leq K \leq 25$ -re legyen:
>
> $$e_K(x) = x + K \bmod 26 \text{ és}$$
> $$d_K(y) = y - K \bmod 26.$$

Sok más hasonló elven mûködõ rendszer ismeretes, például a helyettesítéses, periodikus helyettesítéses és kulcsfolyamos rejtjelezés. Ezen kódolási módszerek feltöréséhez nem kell semmi m tenni csak meg kell figyelni az adott nyelvben az egyes betûk elõfordulásának gyakoriságát és ezt összehasonlítva a titkosított szövegben lévõ betûk gyakoriságával. (Nincs azonban szerencsénk, ha a kódolt üzenet rövid.)

3.2.2. Hill módszer

1929-ben Lester S. Hill által megalkotott titkosítási módszer lényege a következõ: adott m pozití egész. Az üzenetet m hosszú részekre vágva, az eredeti karakterek helyett az üzenet m hosszú részei mint vektortér felett értelmezett, lineáris kombinációt küldünk el. A megfejtés hasonló módon történ kódolt üzenet m hosszú részeire ismét lineáris kombinációt alkalmazunk, hogy visszanyerjük az ered szöveget.

A **Hill** módszer formálisan definiálva:

> Legyen m rögzített, pozitív egész. Legyen $\mathcal{P} = \mathcal{C} = (\mathbf{Z}_{26})^m$ illetve $\mathcal{K} = \{m \times m$ invertálható mátrixok $\mathbf{Z}_{26}$ felett$\}$. Tetszõleges K kulcsra legyen:
>
> $$e_K(x) = xK \text{ és}$$
> $$d_K(y) = yK^{-1},$$
>
> ahol minden mûvelet $\mathbf{Z}_{26}$ felett értendõ.

A formális definició után nézzünk egy példát:

Legyen $m = 2$. Ebben az esetben az eredeti szöveg és a titkosított szöveg is betû párokból fog álln azaz $x = (x_1, x_2)$ és $y = (y_1, y_2)$. Választunk még egy kulcsot is, amely egy lineáris kombináció lesz, ı hogy: $y_1 = 5x_1 + 12x_2$; $y_2 = 13x_1 + 7x_2$. Természetesen mátrixok segítségével is megfogalmazhatjuk az összefüggést: $(y_1, y_2) = (x_1, x_2)\begin{pmatrix} 5 & 13 \\ 12 & 7 \end{pmatrix}$.

Ezek után meg kell határozni a mátrix inverzét, ahhoz, hogy dekódolni tudjuk az üzenetet. A mátrix inverze, mod 26: $\begin{pmatrix} 21 & 13 \\ 16 & 15 \end{pmatrix}$.

Ellenõrzésként szorozzuk össze a két mátrixot:

$$\begin{pmatrix} 5 & 13 \\ 12 & 7 \end{pmatrix} * \begin{pmatrix} 21 & 13 \\ 16 & 15 \end{pmatrix} = \begin{pmatrix} 5*21+16*13 & 5*13+13*15 \\ 5*21+13*16 & 12*13+7*15 \end{pmatrix} = \begin{pmatrix} 313 & 260 \\ 364 & 261 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \bmod 26.$$

A mátrixszorzás asszociativitása miatt: $y * M^{-1} = (x * M) * M^{-1} = x * (M * M^{-1}) = x * E_m = x$, ahol jelöli a $m \times m$-es egységmátrixot. Így valóban $(y_1, y_2)\begin{pmatrix} 21 & 13 \\ 16 & 15 \end{pmatrix} = (x_1, x_2)$, azaz dekódoláskor az inv mátrixszal való beszorzással visszakapjuk az eredeti szöveget.

Legyen a *teke* szó, amit titkosítani szeretnénk. Két részre bontva: (20,5), amely megfelel a *te* részletnek és (11,5), amely a *ke* részletnek felel meg. Kulcsként választjuk a $K = \begin{pmatrix} 5 & 13 \\ 12 & 7 \end{pmatrix}$ mátrixot illetve dekódoláshoz használjuk az inverzét: $K^{-1} = \begin{pmatrix} 21 & 13 \\ 16 & 15 \end{pmatrix}$. Ezt követõen elvégezve a $(20,5)\begin{pmatrix} 5 & 13 \\ 12 & 7 \end{pmatrix} = (9,15)$ és a $(11,5)\begin{pmatrix} 5 & 13 \\ 12 & 7 \end{pmatrix} = (16,11)$ mátrixszorzásokat megkapjuk a *teke* szór megfelelõ kódolt üzenetet: *iopk*. Dekódolásnál nincs más teendõ, csak el kell végezni a következõ kı mûveletet: $(9,15)\begin{pmatrix} 21 & 13 \\ 16 & 15 \end{pmatrix} = (20,5)$ és a $(16,11)\begin{pmatrix} 21 & 13 \\ 16 & 15 \end{pmatrix} = (11,5)$.
Így helyesen megkaptuk az eredeti üzenetet.

Megjegyzés: a Hill módszer speciális eseteként megkaphatunk egy sokkal régebb óta ismert ún. permutációs módszert is. Ha speciálisan K-nak egy $m \times m$ -es permutáció mátrixot választunk, akkoı kapjuk meg ezt az eljárást.

3.2.3. Data Encryption Standard (DES)

Az IBM cég által kifejlesztett és LUCIFER névre keresztelt titkosítási eljárás továbbfejlesztettje a **DES** névre hallgató, és 1977. január 15-én az amerikai Nemzeti Szabványügyi Hivatal (NBS) által szabványként bejegyzett módszer. Leegyszerûsítve a **DES** az eredeti szöveg 64 bites blokkjaiból, 56 bites kulcsot használva, állítja elõ a 64 bites kódolt üzenetet. Az erõs **DES** algoritmusa a következõ három lépésbõl áll:

- Legyen x a nyílt szöveg egy 64 bites része. Elõször ezt a x bitsorozatot permutálják egy *IP* permutáció mátrixszal. Az eredmény: $x_0 = IP(x)$. Ezt követõen az x_0 bitsorozatot felbontják a felsõ 32 bitet tartalmazó L_0 és az alsó 32 bitet tartalmazó R_0 részre.
- A következõ lépésben egy 16 ismétléses ciklusban a következõ mûveleteket végzik L_i és R_i bithalmazzal minden $1 \le i \le 16$ -ra: $L_i = R_{i-1}$ és $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, ahol $\oplus$ a bitenkéı kizáró vagy mûveletét jelöli, az *f* egy jól meghatározott függvény, amit késõbb részletezek illeı K_i a *K* kulcs által meghatározott 48 bit hosszú bitsorozat.
- Végül $R_{16}L_{16}$ -ra alkalmazva *IP* inverzét, megkapjuk a kódolt üzenetet: $y = IP^{-1}(R_{16}L_{16})$.

Az algoritmus egy lépésének ábrája és az *IP* illetve az IP^{-1} bitpermutáló mátrix megtalálható a 7.3. é 7.4. Függelékben.

Ezek után nézzük meg, hogyan néz ki az *f* függvény. A két bemeneti paramétere közül az elsõ, jelölj *A*-val, 32 bit hosszú; a második paraméter, amit *J*-vel jelölünk 48 bit hosszú. Az *f* mûködése 4 lépésı bontható:

- Legelõször az *A* 32 bites bemenetet kiterjesztik egy *E* kiterjesztési függvénnyel 48 bitre. Így *E* tartalmazza az *A* 32 bitjét meghatározott módon összekeverve 16 olyan bittel, amely kétszer is megjelenik.
- Ezt követõen meghatározzák a $E(A) \oplus J = B$ érték, amelyet nyolc 6 bites részre vágnak: *B* $B_1B_2B_3B_4B_5B_6B_7B_8$.
- Ebben a lépésben nyolc *S*-mátrixot használnak, melyek mindegyike 4 x 16-os és elemei [0,...,1 egészek. Az adott hat bit hosszúságú, mondjuk $B_j = b_1b_2b_3b_4b_5b_6$ -ra meghatározzák az $S_j(B_j$ értékeket a következõképpen: a b_1b_6 bitek binárisan kódolják az *r*. sorát az S_j-nek (r = 0..3 -rı a középsõ bitek: $b_2b_3b_4b_5$, binárisan kódolják a *c*. oszlopát az S_j-nek (r = 0..15 -re). Ezután S_j értéke $S_j(r, c)$ lesz, amely négy bit hosszú. Ilyen módon történik $C_j = S_j(B_j)$ értékének meghatározása $1 \le j \le 8$ -ra.
- A $C = C_1C_2C_3C_4C_5C_6C_7C_8$ bitsorozaton alkalmazunk egy *P* permutációt, és az így kapott *P*(lesz a kimenete az *f* függvénynek: $f(A, J)$.

Az *E* kiterjeszési függvény, az *f* függvény és az *S*-mátrixok és a *P* permutáció megtalálhatók a 7.5., 7 és a 7.8. Függelékben.

Végül nézzük meg, hogy miként áll elõ az 56 bites kulcsból a kódolás második lépéséhez szükséges kulcstáblázat. Az 56 bites kulcsot elõször is kiegészítik nyolc paritásbittel, amely az esetleges hibák javítására szolgál. Az így kapott 64 bites *K* kulcsból számítják a K_i kulcstáblázatokat úgy, hogy az az elõbb említett paritásbitek értékétõl független legyen.

- A *K* kulcs paritásbitek nélküli 56 bites részét permutálják egy *PC*-1 rögzített mátrixszal. A kap eredményt *PC*-1(*K*)-t felbontják a felsõ 28 bitet tartalmazó C_0 és az alsó 28 bitet tartalmazó *D* részre.
- Minden $1 \le i \le 16$ -ra kiszámításra kerülnek a $C_i = LS_i(C_{i-1})$, $D_i = LS_i(D_{i-1})$ és a $K_i = PC\text{-}2$ (C_iD_i) értékek, ahol LS_i egy bitléptetõ függvény, amely jobbról balra lépteti a biteket *i* értékétı

függõen eggyel, ha $i = 1, 2, 9, 19$ és kettõvel egyébként.

A végeredményként kapott kulcstáblázat illetve a *PC*-1 és *PC*-2 permutációk megtalálható a 7.9. és 7.10. Függelékben.

A dekódolás algoritmusa lényegében megegyezik a titkosítás algoritmusával, azzal a különbséggel, h a bemenete természetesen az y kódolt üzenet és a dekódolás alkalmával a most említett K_i kulcstáblázatokat ellentétes irányban kell használni, azaz elõször a K_{16}, K_{15}, ..., majd a K_1 -et.

3.2.4. A titkos kulcsú kriptorendszerek gyengéje

Mi is az, amiért az üzeneteinket kódolni szeretnénk? Az a tény, hogy nem biztonságos a csatorna, amelyen keresztül kommunikálni szeretnénk. Minden titkos kulcsú rendszer azonban abból indul ki, hogy a kommunikáló felek megegyeztek abban, hogy milyen kulccsal is kódolják üzeneteiket. De ezt hogyan tették meg? Nem feltétlenül van lehetõségük felkeresni egymást és megbeszélni, hogy milyen kulcsot is használjanak a titkosítás során. Akkor pedig a kulcsot is el kell küldeniük egymásnak. Enn problémának a megoldása újabb kutatásokhoz vezetett. Készültek úgynevezett biztonságos *kulcskio:* algoritmusok, amelyek a kommunikáló felek közötti kulcscserét tették lehetõvé, de hamarosan egy forradalmian új alapgondolat is megszületett, amelyre kriptorendszerek egy teljes családja épül.

3.2.5. A Diffie-Hellman kulcskiosztó algoritmus

Tegyük fel, hogy Alice és Bob szeretnének valamelyik titkos kulcsú kriptorendszer által kódolt üzenetet váltani. Ekkor meg kell egyezniük egy közös, titkos kulcsban. A következõ módon tehetik : ezt:

A rendszert két nyilvános paraméter határozza meg: egy p prím és egy $g \in \mathbf{Z}_p$ primitív elem. Legelõször Alice és Bob generál egy-egy véletlen értéket: a-t és b-t. A két nyilvános paraméter segítségével meghatározzák a $g^a \bmod p$ és $g^b \bmod p$ értékeket, és el is küldik azokat egymásnak. Vé Alice meghatározza a $k_{ab} = (g^b)^a \bmod p$ értéket illetve Bob kiszámolja a $k_{ba} = (g^a)^b \bmod p$ értéket. Mivel, $k_{ab} = k_{ba} = k$ ezért mindketten rendelkeznek ugyanazzal a titkos kulcsértékkel, amelyet a kód során használhatnak.

A **Diffie-Hellman** kulcskiosztó algoritmus formális leírása:

Legyen $p \geq 3$ prím és $g \in \mathbf{Z}_p$ primitív elem. Tetszõleges $a,b \in \mathbf{Z}$ értékekre határozzuk meg a:

$$k_a = g^a \bmod p \text{ és}$$
$$k_b = g^b \bmod p \text{ értékeket.}$$

A közös titkos kulcs a $k = k_{ab} = (g^b)^a = k_{ba} = (g^a)^b$ lesz.

4. Nyilvános kulcsú kriptorendszerek

4.1. A nyilvános kulcsú kritorendszerek alapgondolata

Az 1976-ban Whitfield Diffie és Martin E. Hellman által kigondolt elvek új korszakot nyitottak az adatbiztonság és kódolás történetében. Az eddig használatos, és az elõzõ fejezetben bemutatott, titk kulcsú kriptorendszerek hibáit kijavítva egy teljesen új ötlettel álltak elõ. Az ötlet a következõ volt: olyan kulcspárt használni a kódolás és a dekódolás alkalmával, amelynek az egyik része egy mindenk által ismert nyilvános kulcs, a másik része viszont csak a címzett által ismert titkos kulcs. Az is fonto alapkövetelmény volt, hogy a nyilvános kulcs ismeretében ne lehessen meghatározni a dekódolásnál használatos privát kulcsot. Ezek után nézzük meg, hogy miként is néz ki az üzenet továbbítása:

A kódoláshoz szükséges kulcs két részbõl áll: egy nyilvános E rejtõ és egy titkos D fejtõ részbõl. Minden, a kommunikációban résztvevõ félnek, van egy-egy ilyen kulcs párja. Követelmény az E, D kulcs párra, hogy egymás után alkalmazva visszaadják az eredeti értéket, azaz $D(E(x)) = E(D(x)) =$. Így amikor Alice az x üzenetet el akarja küldeni Bobnak, az üzenet helyett Bob nyilvános kulcsával titkosított üzenetet küldi el, azaz $E_b(x)$-et, amit csak Bob tud megfejteni, úgy hogy alkalmazza a sajá fejtõkulcsát: $D_b(E_b(x)) = x$.

4.2. Üzenet hitelesítés: digitális kézjegy

Node Bob honnan tudja, hogy valóban Alice üzenetét kapta meg? Mivel a kulcspár E titkosító rés nyilvános, az üzenetet elvileg bárki küldhette. Az lenne a megoldás, ha Alice „aláírásával" hitelesíten üzenetet, azaz az üzenetbõl egyértelmûen kiderülne, hogy azt õ küldte. Erre a kérdésre is nagyon egyszerû megoldás adódik. Alice a kódolt üzenetet $E_b(x)$-et a saját, titkos fejtõ kulcsával is kódolja, amit csak õ ismer. Azaz, ha bárki más Alice nevében kívánna üzenetet küldeni, akkor szüksége lenne Alice titkos fejtõ kulcsára, ahhoz, hogy azt a látszatot keltse, hogy Alice küldte az üzenetet. Így az üzenet $D_a(E_b(x))$ lesz. Ezt megkapva Bob elõször alkalmazza Alice nyilvános rejtõ kulcsát: $E_a(D_a(E(x))) = E_b(x)$, majd a saját fejtõkulcsát: $D_b(E_b(x)) = x$. Ezzel visszakapja az eredeti üzenetet.

Abban az esetben, ha egy harmadik személy próbálja megfejteni az üzenetet, akkor elõször õ is alkalmazza Alice nyilvános rejtõ kulcsát, így megkapja $E_b(x)$-et, de mivel nem ismeri Bob titkos fejtõ kulcsát, ezért nem tud hozzájutni az üzenet tartalmához.

Ezek után lássuk néhány matematikai módszert, amelyeket a fenti gondolatmenet implementálására dolgoztak ki:

4.3. Az RSA algoritmus

A módszer elnevezése megalkotóik vezetékneveinek kezdõbetûibõl áll össze: **Rivest**, **Shamir** és **Adleman**. Az általuk kidolgozott séma arra a tényre épül, hogy jelenleg nem ismeretes hatékony, gy eljárás egész számok faktorizációjára.

Az **RSA** formális leírása:

Legyen $n = pq$, ahol $p,q \geq 3$ prímek. Továbbá legyen $\mathcal{P} = \mathcal{C} = \mathbf{Z}_n^*$ illetve $\mathcal{K} = \{(n, p, q, a, b)$ olyan, hogy $n = pq$, p, q prím, $ab \equiv 1 \mod \phi(n)\}$. Tetszõleges $K = (n, p, q, a, b)$ kulcsra definiáljuk:

$$e_K(x) = x^b \bmod n \text{ és}$$
$$d_K(y) = y^a \bmod n,$$

ahol $x, y \in \mathbf{Z}_n^*$. Az n és a b értékek nyilvánosak, a p, q, a értékek pedig titkosak.

Ezek után nézzük meg miért is mûködik az eljárás:

A számításokat $\mathbf{Z}_n$ gyûrûben hajtjuk végre, ahol $n = pq$, ahol p, q két páratlan prím. Ilyen n-re $\phi($
$(p - 1)(q - 1)$. Ellenõrizzük, hogy a kódoló és a dekódoló függvény valóban egymás inverzei.
Mivel $ab \equiv 1$ mod $\phi(n)$, ezért $ab = t\phi(n) + 1$, valamilyen $t \geq 1$ egészre. Legyen $x \in \mathbf{Z}_n^*$. Így: $(x^b$
$a \equiv x^{t\phi(n)+1} \equiv (x^{\phi(n)})^t x \equiv 1^t x \equiv x$ mod n. Ezzel beláttuk, hogy egymás után alkalmazva a kódoló és dekódoló függvényeket, az eredeti szöveget kapjuk vissza.

Szemléltetésként nézzünk meg egy egyszerû példát:

Tegyük fel, hogy Bob a $p = 101$ és $q = 113$ prímeket választotta. Ekkor $n = 11413$ és $\phi(n) = 1120$ adódik. Akkor és csak akkor tudja használni az eljárásban b-t kitevõnek, ha b nem osztható $\phi(n)$ egyetlen prímtényezõjével sem, azaz lnko(b, $\phi(n)$) = 1. (Ez annak a feltétele, hogy b-nek létezzen multiplikatív inverze mod $\phi(n)$.) Bob a $b = 3533$-t választja. Ezek után a kiterjesztett euklideszi algoritmussal meghatározza a 3533 multiplikatív inverzét modulo 11200. Adódik: $b^{-1} = 6597$ mod 11200. Ezzel megkapta a titkos megfejtõ kulcsot. Ezek után Bob közzéteszi kulcsának nyilvános rés $n = 11413$ és $b = 3533$.

Tegyük fel, hogy Alice 9726 nyílt szöveget akarja elküldeni Bobnak. Elõször meghatározza a 97263533 mod 11413 = 5761 értéket és azután elküldi Bobnak.

Mikor Bob megkapja a 5761 kódot, kiszámítja a 5761^{6597} mod 11413 = 9726 értéket, és ezzel visszakapja az eredeti üzenetet.

4.3.1. Az RSA biztonsága

Az RSA algoritmus biztonsága azon múlik, hogy a kódoló függvény $e_K(x) = x^b$ mod n egyirányú, tehát az illetéktelenek számára a dekódolása igen hosszú idõt vesz igénybe. A megoldás kulcsa abbaı rejlik, hogy valaki n ismeretében elõ tudja-e állítani $\phi(n)$-et. Ugyanis a nem más, mint b multiplikatív inverze modulo $\phi(n)$, amit a kiterjesztett euklideszi algoritmussal, $\phi(n)$ értékét ismerve, könnyen meg lehet határozni. Azonban jelenlegi tudásunk szerint n értékébõl $\phi(n)$ kiszámítása csak úgy lehetséges, hogy elõször n-et prímtényezõkre bontjuk. (A kriptográfiai kutatások egyik jelenlegi legfontosabb problémája annak igazolása, vagy cáfolása, hogy az RSA feltörése vajon ekvivalens-e a modulus faktorizációjával. Ez a probléma viszont még megoldatlan.) Ugyancsak nem ismert gyors, polinomiál idejû algoritmus n faktorizációjára. Ez azt jelenti, hogy a egy 10^{12}-en nagyságrendû számot még *kényelmesen* prímtényezõkre tudunk bontani, de 10^{130}-on nagyságrendû számok felbontása, legalábl módszer megalkotása 1976. táján, az akkori technika és algoritmusok eszközeivel ~10^{20} évet vett vc igénybe. Tehát, ha elég nagy p, q értéket választunk, akkor elmondhatjuk, hogy elég biztonságos a rendszer. (Számos hardware megoldás támogatja az 512 bites RSA-t, amely 155 jegyû - tízes számrendszerbeli - számnak felel meg, de úgy tûnik, hogy a mai boszorkányos ügyességû algoritmus és a villámgyors technika már képesek egy ekkora szám felbontására. Ugyanis 1999. augusztus 22-éı Herman te Riele által vezetett, hat nemzet legkiválóbb matematikusaiból álló, kutatócsoportnak hét hónapos számolást követõen sikerült egy nehéznek tekintett 512 bites számot faktorizálnia. [9],[10])

A 7.13. Függelékben a faktorizálás technikai részletei találhatók meg.

4.3.2. Az RSA megvalósítása

Nézzük meg mit is kell tennie Bobnak, ha az RSA kriptorendszert akarja használni:

1. elõ kell állítani két nagy prímet: p, q -t (körülbelül 100 jegy nagyságrendût),
2. meg kell határoznia $n = pq$, $\phi(n) = (p - 1)(q - 1)$ értékeket,
3. választania kell egy véletlen $1 < b < \phi(n)$ értéket úgy, hogy lnko$(b, \phi(n)) = 1$,
4. az euklideszi algoritmussal meg kell határoznia $a = b - 1 \bmod \phi(n)$ értékét és
5. közzé kell tennie n illetve b értékét.

A fenti listából csak az elsõ lépés, ami egy kis problémát jelent: kell találni két körülbelül 100 jegy prímet. Honnan lehet ilyet szerezni? A helyzet az az, hogy ezeket a prímeket mindenkinek magának l legyártania, mert ha valakitõl szívességbõl vagy pénzért megkapná, akkor a biztonságát adná cserébe prímekért. A prímek keresése egy elég összetett feladat, amelyet a következõ fejezetben részletezek.

4.4. Nagy prímek keresése, prímtesztek

A nagy prímszámtétel miatt, ha k bites prímeket keresünk, akkor annak a valószínûsége, hogy egy véletlenül választott páratlan egész prím ~ c/k. Tehát, elég hatékony, ha úgy keresünk prímeket, hog egy megfelelõ méretû páratlan egészet választunk véletlenszerûen és utána elkezdjük tesztelni, hogy valóban prím-e. Az alábbiakban definiálunk egy ilyen ún. prímtesztelõ algoritmust:

4.4.1. A Solovay-Strassen prímteszt

Legyen $n > 2$ páratlan szám. Kérdés n prím-e?

1. Válasszunk egy véletlen $1 \leq c < n$ egészet.

Ha $$\text{lnko}(c,n) = \begin{cases} n, \text{ akkor válasszunk új } c\text{-t;} \\ 1 < d < n, \text{ akkor } n \text{ összetett;} \\ 1, \text{ akkor lépjünk a 2. lépésre.} \end{cases}$$

2. Ha $\left(\frac{c}{n}\right)_J \equiv c^{(n-1)/2}$ mod n, akkor n prím, különben n összetett.
3. Ismételjük az eljárást megfelelõen sokszor.

Említsünk meg egy tételt (bizonyítás nélkül), amellyel a teszt helyességét tudjuk ellenõrizni:

Tétel: Ha n páratlan összetett szám, akkor létezik c egész úgy, hogy lnko$(c, n) = 1$ és $\left(\frac{c}{n}\right)_J \not\equiv c^{(n-1}$ mod n.

Következmény: Tegyük fel, hogy n páratlan összetett szám. Legyen C_1 = {olyan c-k, amelyekre lnl $(c, n) = 1$ és teljesül a $\left(\frac{c}{n}\right)_J \equiv c^{(n-1)/2}$ mod n kongruencia} és legyen C_2 = {olyan c-k, amelyekre ln $(c, n) = 1$ és nem teljesül a $\left(\frac{c}{n}\right)_J \equiv c^{(n-1)/2}$ mod n kongruencia}. Ekkor $|C_1| \leq |C_2|$.

ui: Legyen $c \in C_1$ és $c' \in C_2$. Ekkor *cc'*-re nem fog teljesülni a kongruencia, ezért $|C_1| \leq |C_2|$.

Elnevezés: A C_2 halmazt szokás „tanú" halmaznak is hívni, hiszen a benne szereplõ elemek tanuk az összetettségére.

4.4.1. A Solovay-Strassen módszer hatékonysága

Abban az esetben, ha azt az eredményt kapjuk, hogy a kérdéses n páratlan szám összetett, akkor biztosak lehetünk benne, hogy összetett. Mi a helyzet az n prím válasszal? Tegyük fel, hogy az eljárá *m*-szer ismételtük, és minden esetben azt a választ kaptuk, hogy *n* prím. Legyen B esemény, hogy *n* összetett, és legyen A esemény, hogy *n*-et prímnek nyilvánítjuk. Ekkor a teszt egyszeri végrehajtásáv (A | B) $\leq 2^{-1}$. Így, ha az eljárást *m*-szer futtatjuk, akkor P(A | B) $c \in C_1$ 2^{-m} fog teljesülni, azaz *m* értékének növelésével tetszõlegesen pontossá tehetõ az eredmény, de mindig marad 2^{-m} bizonytalan:

Egy hatékonyabb módszert is szeretnék megemlíteni a **Miller-Rabin** prímtesztet. Azért használjál gyakorlatban szívesebben, mert gyorsabb mint az elõbb tárgyalt **Solovay-Strassen** teszt és a hatékonyságai is jobb, mivel a „tanú" halmaza bõvebb. A teszt Mathematica® kódja megtalálható a 7 Függelékben.

4.5. Az ElGamal kriptorendszer

A rendszer a **diszkrét logaritmus** néven ismert problémán alapszik. A következõrõl van szó:

4.5.1. Diszkrét logaritmus probléma

Legyen adott $I = (p, \alpha, \beta)$ hármas, ahol *p* prím, $\alpha \in \mathbf{Z}_p^*$ primitív gyök illetve $\beta \in \mathbf{Z}_p^*$.   Keressük azt az *a* egészet, amelyre $0 \leq a \leq p-2$ és $\alpha^a \equiv \beta \bmod p$.

A probléma megfelelõen bonyolult, ha gondosan választjuk meg *p* értékét. Ugyanis nincs ismert polinomiális idejû algoritmus a **diszkrét logaritmus** probléma megoldására. Az ismeretes támadási lehetõségek kockázatát csökkentjük, ha jól választjuk meg *p* értékét. Legyen *p* legalább150 jegyû úg hogy (*p* - 1)-nek legyen legalább egy „nagy" prímtényezõje. A diszkrét logaritmus probléma hasznos abban rejlik, hogy megtalálni a diszkrét logaritmust (eddig úgy tûnik) nehéz, viszont az inverz mûvel a hatványozás, könnyen és gyorsan számítható az ismételt négyzetre emelés módszerrel. Más szóval: modulo *p* hatványozás megfelelõ *p* prímre egyirányú függvény.

Megjegyzés: A kriptográfiai kutatások egy másik igen fontos témája: az elõzõ fejezetben tárgyalt a **Diffie-Hellmankulcskiosztó algoritmust** feltörni és a **diszkrét logaritmus problémát** megoldani ekvivalens-e vagy sem. Ueli M. Maurer és Stefan Wolf 1996. április 18-án publikált közös cikkükben igen fontos lépéseket tettek annak irányában, hogy igazolják a két probléma ekvivalenciáját bizonyos feltételek teljesülésével [7].

4.5.2. Az ElGamal módszer

Legyen p olyan, hogy a $\mathbf{Z}_p$ felett vett diszkrét logaritmus probléma nehéz, illetve válasszunk egy $\alpha \in \mathbf{Z}_p$ primitív gyököt. Legyen továbbá $\mathcal{P} = \mathbf{Z}_n^*$, $C = \mathbf{Z}_n^* \times \mathbf{Z}_n^*$, illetve $\mathcal{K} = \{(p, \alpha, a, \beta)$ úgy, hogy $\beta \equiv \alpha^a \bmod p\}$. A p, α és a β ,értékek nyilvánosak, az a értéke pedig titkos.
Adott $K = (p, \alpha, a, \beta)$ kulcsra és egy tetszõleges, titkos $k \in \mathbf{Z}_{p-1}$ -re legyen:

$$e_K(x, k) = (y_1, y_2),$$

ahol $y_1 = \alpha^k \bmod p$ illetve $y_2 = x\beta^k \bmod p$ és $y_1, y_2 \in \mathbf{Z}_p^*$ -ra pedig:

$$d_K(y_1, y_2) = y_2(y_1^{a})^{-1} \bmod p.$$

Megjegyzés: Vegyük észre, hogy az **ElGamal** titkosítási rendszer nem determinisztikus, azaz a kód üzenet mind a nyílt szövegtõl x, mind Alice által véletlenszerûen választott k értéktõl is függ.

Hétköznapi nyelven megfogalmazva az **ElGamal** módszer mûködését: az adott nyílt szöveget „maszkoljuk" azzal, hogy megszorozzuk β^k-val, így megkapjuk y_2-t. Az α^k értéket ugyancsak továbbítjuk a kódolt üzenetben. Bob, aki ismeri a titkos kulcsot a-t, meg tudja határozni β^k-t α^k értékébõl, és így „el tudja távolítani a maszkot", azáltal, hogy y_2-t osztja β^k-val, és így megkapja az eredeti üzenetet.

Nézzünk egy példát:

Tegyük fel, hogy $p = 2579$, $\alpha = 2$ és $a = 765$. Adódik tehát, hogy $\beta = 2^{765} \bmod 2579 = 949$. Mos tegyük fel, hogy Alice $x = 1299$ üzenetet akarja elküldeni Bobnak, illetve azt, hogy az általa véletlenszerûen választott titkos szám a 853. Ezek után meghatározza az $y_1 = 2853 \bmod 2579 = 43$: az $y_2 = 1299 * 949^{853} \bmod 2579 = 2396$ értékeket és elküldi Bobnak az $y = (y_1, y_2)$ párt.

Miután Bob megkapta az $y = (435, 2396)$ párt kiszámolja az $x = 2396 * (435^{765})^{-1} \bmod 2579 = 1$ értéket, és ezzel visszakapja az eredeti üzenetet.

4.5.3. Algoritmusok diszkrét logaritmus problémára

Mivel dolgozatom fókuszában a kriptorendszerek és a hozzájuk kapcsolódó algebrai elmélet áll, íg csak megemlíteném néhány, a diszkrét logaritmus problémára készült algoritmust, amelyeket az ezen alapon mûködõ kriptorendszerek feltörésére lehet használni. Ez a téma, a kriptorendszerek feltörése, önmagában elég érdekes és önálló szakdolgozati témaként is megállná a helyét, ezért is csak megemlítem ezen módszereket, jelezve a lehetõségét az esetleges támadási felületeknek.
Három, a diszkrét logaritmus probléma megoldására készült algoritmust említenék meg: a **Shanks** algoritmust, a **Pohling-Hellman** algoritmust és a leghatékonyabbnak talált **Index Kalkulus** eljárást.

5. Az elliptikus görbék áttekintése

5.1. A harmadfokú görbékrõl

Ebben a fejezetben használni fogjuk az algebrai geometria néhány alapfogalmát, mint például projekt síkgörbe, homogén koordinátarendszer, projektív koordinátarendszer-transzformáció stb.

Def: Legyen T test. Egy általános kétváltozós harmadfokú síkgörbe egyenlete: $ax^3+bx^2y+cxy^2+dy^3+ex^2+fxy+gy^2+hx+iy+j=0$ alakú, ahol $a,b,c,d,e,f,g,h,i,j\in T$.

Egy igen általános tételt említsünk meg, amelyre a késõbbiekben lesz szükségünk:

Bézout-tétel: Egy m-ed és egy n-ed fokú projektív görbének multiplicitással számolva összesen $m*n$ metszéspontja van.

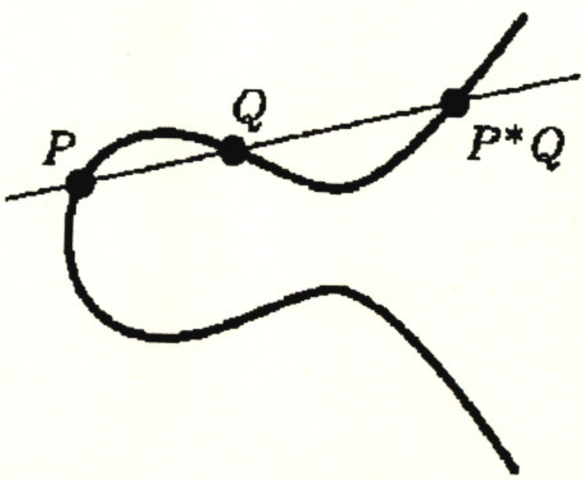
Megjegyzés: Számunkra speciálisan két harmadfokú görbére vonatkoztatva használjuk, tehát két harmadfokú görbének kilenc metszéspontja van.

Chasles-tétel: Legyen adott C, C_1, C_2 három harmadfokú görbe. Ha C áthalad C_1 és C_2 kilenc metszéspontja közül nyolcon, akkor C-nek át kell haladnia a kilencedik metszésponton is.

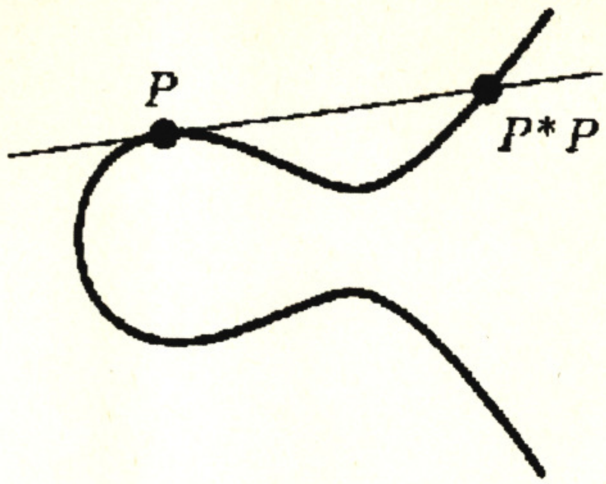
A bizonyítást nem részletezzük ([3] szép elemi bizonyítást ad). Kiderül, hogy az adott nyolc pontc átmenõ görbék 2-dimenziós vektorteret határoznak meg. Ha F_i jelöli C_i egyenletét, akkor a nyolc ponton átmenõ görbék egyenletei éppen $\lambda_1 F_1+\lambda_2 F_2=0$ alakúak. Ennek az egyenletnek pedig nyilvánvalóan megoldása a kilencedik metszéspont is.

Célunk az lenne, hogy a görbe pontjaiból csoportot hozzunk létre. Ennek érdekében definiálnunk egy mûveletet a görbe pontjain. Tegyük fel tehát, hogy egy olyan harmadfokú síkgörbénk van, amely minden pontjához egyértelmûen húzható érintõ, beleértve az esetleges végtelen távoli pontot is.

Egy kézenfekvõ ötlet: a görbe adott két pontjához rendeljük hozzá az általuk meghatározott egyen és a görbe harmadik metszéspontját! Az adott P és Q pontokhoz rendeljük hozzá a $P*Q$ pontot. Szemléletesen valahogy így:

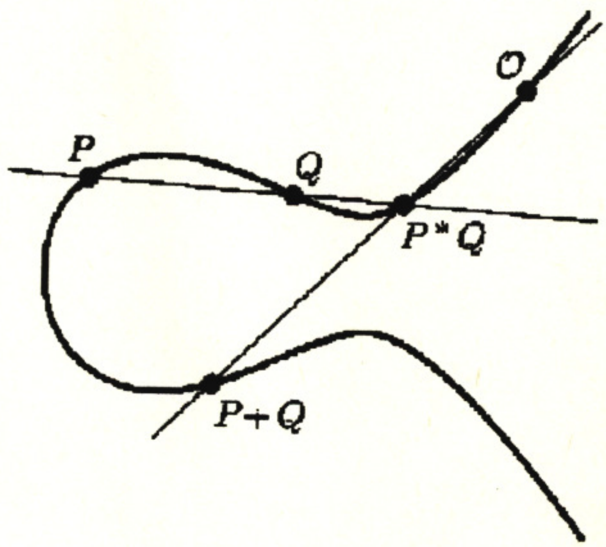


Hasonlóan meg lehet kapni $P*P$ értékét, ha P-beli érintõ és a görbe metszéspontját vesszük.

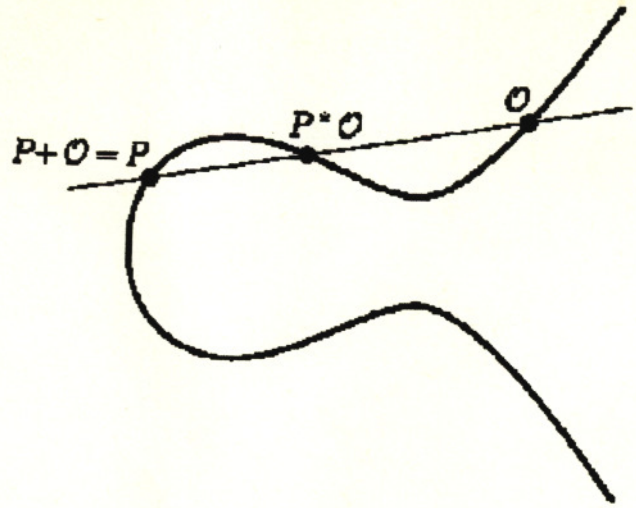


Sajnos azt kell megállapítanunk, hogy ez a *mûvelet* annak ellenére, hogy elég kézenfekvõnek tûnik nem elégíti ki igényeinket, ugyanis a görbe pontjain értelmezett '*' mûveletre nem alkotnak csoporto görbe pontjai. Könnyen látható, hogy nincs egységelem. Ezért meg kell változtatnunk a mûvelet definícióját, hogy eleget tegyen elvárásainknak.

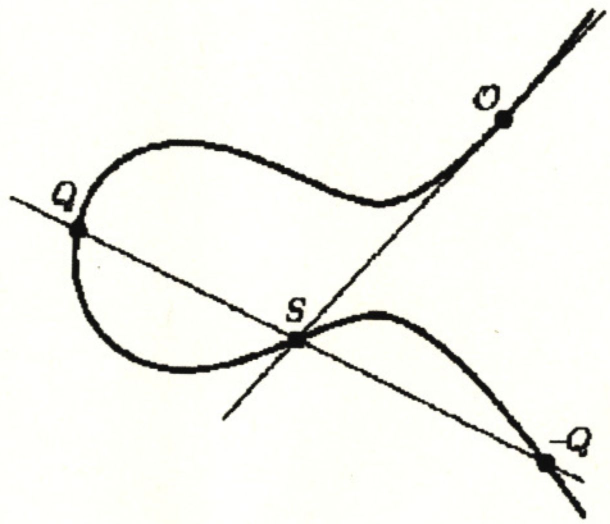
Válasszuk ki tetszõlegesen a görbe egy pontját, O-t, amely a csoport null eleme lesz. (A mûveletet jelöljük '+' -szal, mert egy kommutatív csoportot fogunk kapni.) Ezek után P és Q összegekén értsü következõket: P, Q egyenesének harmadik metszéspontja a görbével legyen $P*Q$, ezt a pontot összekötve O-val és a harmadik metszéspontot véve megkapjuk $P+Q$-t. Így definíció szerint: $P+Q=O*(P*Q)$. Szemléletesen:



Látszik, hogy a mûvelet kommutatív, azaz $P+Q = Q+P$. Most ellenõrizzük, hogy O valóban null elem-e, azaz $P+O = P$. Miért is van ez így? O, P összekötõ egyenesének harmadik metszéspontja a görbével $P*Q$. Ezt összekötve O-val, a harmadik metszéspont éppen P lesz. Így valóban $P+O = P$. Ahogyan az ábra mutatja:

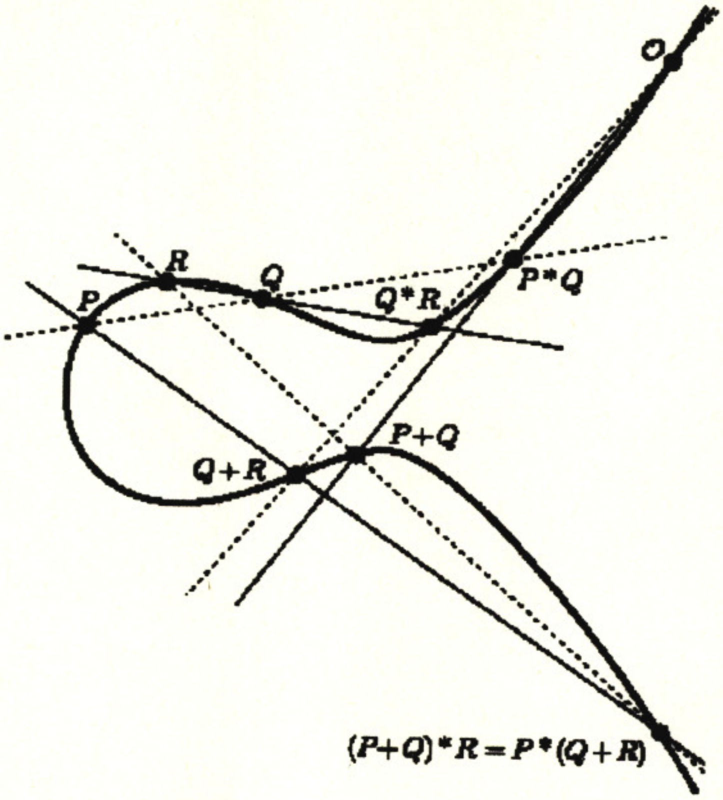


Most megmutatjuk, hogy $\forall Q$ görbe ponthoz $\exists -Q$ pont úgy, hogy $Q+(-Q)=O$. Ahhoz, hogy belássuk, O-ból húzzunk érintõt a görbéhez (feltéve, hogy a görbe nem szinguláris, tehát érintõ húzh minden pontba). Az érintõ és a görbe másik metszéspontját jelöljük S-sel. Q-t S-sel összekötve, a harmadik metszéspont $Q*S$, amely megfelel $-Q$-nak. Ellenõrzésként, ha Q-t és $-Q$-t összeadjuk, az általuk meghatározott egyenes S-ben metsz a görbét, S-et O-val összekötve a harmadik metszéspont kapjuk $Q+(-Q)$-t. $\overline{OS}$ egyenes O-beli érintõ, az O pont multiplicitása kettõ, azaz a harmadik metszéspont ugyancsak O. Így valóban: $Q+(-Q)=O$. Ahogyan az ábra mutatja:



Ahhoz, hogy ténylegesen csoportot kapjunk már csak a mûvelet asszociativitását kell belátnunk. Ennek az állításnak a teljes általánosságában való bizonyítása, pontok esetleges egybeesésekor igen bonyolult technikákat és számolást igényel, ezért a teljes általánosságtól eltekintünk és csak szemléltetésként adjuk ezt a bizonyítást, arra az esetre, ha nincsenek egybeesések. Legyenek P, Q és görbe pontjai. Azt kell belátnunk, hogy $(P+Q)+R = P+(Q+R)$. Elég belátni, hogy $(P+Q)*R = P*(Q+R)$, ugyanis, ha ezt a pontot O-val összekötjük $(P+Q)+R$-t és $P+(Q+R)$-t kapunk, ami az elõzõ egyenlõség miatt egybe fog esni. Ahhoz, hogy megkapjuk $P+Q$-t össze kell kötni $\overline{PQ}$ egyenes görbével vett harmadik metszéspontját, $P*Q$ pontot-t, O-val. $P+Q$-t összekötve R-rel kapjuk $(P+Q)$ pontot. Hasonlóan, ahhoz, hogy megkapjuk $P*(Q+R)$ pontot elõször össze kell kötnünk $\overline{QR}$ egyen(görbével vett harmadik metszéspontját, $Q*R$ pontot-t, O-val, majd össze kell kötni $Q*R$-t P-vel és a görbével vett harmadik metszéspontként megkapjuk $P*(Q+R)$-t. Vajon P-t és $(Q+R)$-et összekötõ

egyenes illetve R-et és $(P+Q)$-t összekötõ egyenes a görbén metszi egymást? Ha igen, akkor beláttul hogy $(P+Q)*R = P*(Q+R)$.
Tekintsünk az ábrára!



Adott kilenc pont: O, P, Q, R, $P*Q$, $P+Q$, $Q*R$, $Q+R$ és az elõbb említett kérdéses metszéspont. azt jelenti, hogy van két harmadrendû görbénk, amelyek ezen a nyolc ponton mennek keresztül. A k‹ görbe: az eredeti és a három, az ábrán szaggatott vonallal jelzett, egyenes szorzata, amely egy harmadrendû görbe. Mivel nyolc adott ponton átmegy mindkét görbe a **Chasles-tételt** alkalmazva kapjuk, hogy a két görbe áthalad mind a kilenc ponton, azaz a kérdéses metszéspont rajta van a görb tehát: $(P + Q)*R = P*(Q+R)$, így $(P+Q)+R = P+(Q+R)$.

Mi van abban az esetben, ha egy másik pontot választunk ki null elemnek, mondjuk O'-t? Ekkor a elõbbivel izomorf csoportot kapunk a következõ izomorfizmussal: $P \rightarrow P+(O-O')$.

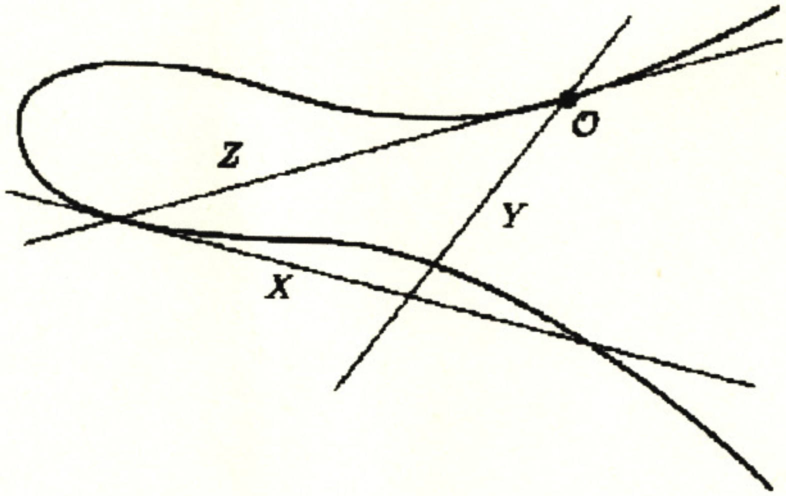
Ezzel beláttuk, hogy a görbe pontjain értelmezett '+ ' mûvelet csoportot alkot a görbe pontjaival.

5.2. Elliptikus görbék Weierstrass-féle normálalakja

Egy a komplex test feletti érveléssel megmutatjuk, hogy miként lehet a görbe egyenletét egyszerûbb alakra hozni annak érdekében, hogy könnyebben tudjunk dolgozni velük. Az ún. Weierstrass-féle no alak, amelyre szeretnénk hozni a görbéket a következõképpen néz ki általánosan: $y^2 = x^3 - g_2x - g$
A továbbiakban mi egy másik ugyancsak Weierstrass alaknak nevezett formába írjuk a görbéket: $y^2 = x^3 + ax^2 + bx + c$ ún. rövid Weierstrass alak.

Megjegyzés: Nem minden esetben lehet ilyen egyszerû alakra hozni tetszõleges test fölött a görbét. Például 2 és 3 karakterisztikájú test fölött nem létezik ilyen egyszerû normálalak, de a dolgozatomba nem használom azokat, ezért nem is térek ki rájuk. Gyakorlati felhasználásban azonban a 2 karakterisztikájú testek igen fontosak. Ott $y^2 + xy = x^3 + ax^2 + b$ alakra hozható az egyenlet.

Megfelelõen megválasztva a projektív koordinátarendszer tengelyeit érjük el, hogy a projektív transzformációt követõen a görbe egyenlete egyszerûsödjék. Legyen *O* a görbe adott pontja azonos azzal, amelyet a '+' mûvelet null eleméül választottunk. Húzzuk meg *O*-ban az érintõt! Ez legyen a projektív *Z*-tengely. Az érintõ és a görbe másik közös pontjából ismét húzzunk érintõt a görbéhez, íg megkapjuk az *X*-tengelyt. Végül *Y*-tengelynek válasszunk bármely *O*-n átmenõ egyenest, amely nem megy át az *X*-tengely és a görbe metszéspontján, illetve különbözik a *Z*-tengelytõl. Mint az ábrán:



Megjegyzés: Abban az esetben, ha *O* inflexiós pont lenne *X*-tengelyként bármely *O*-n át nem menõ egyenest választhatnánk.

Így választva a koordináta tengelyeket és az $x = \frac{X}{Z}, y = \frac{Y}{Z}$ helyettesítést végrehajtva és algebrailag rendezve a görbe egyenletét, a következõ formára hozhatjuk azt: $xy^2 + (ax+b)y = cx^2 + dx + e$.

Ezek után *x*-szel beszorozva kapjuk: $(xy)^2 + (ax+b)xy = cx^3 + dx^2 + ex$.

Majd *xy* helyére ismét *y*-t írva kapjuk: $y^2 + (ax+b)y = c'x^3 + d'x^2 + e'x$.

Egy újabb lineáris helyettesítéssel *y* helyett $y - \frac{1}{2}(ax+b)$-t írva kapjuk: $y^2 =$ *x-ben harmadfokú függvény*.

Elõfordulhat még, hogy az *x-ben harmadfokú függvény* fõegyütthatója nem 1. Ekkor *x*-et λx-szel, *y*-pedig $\lambda^2 y^2$-tel helyettesítve, ahol λ a harmadfokú függvény fõegyütthatója, megkapjuk a kívánt alak(azaz: $y^2 = x^3 + ax^2 + bx + c$.

Def.: A normál formában felírt $y^2 = f(x) = x^3 + ax^2 + bx + c$ harmadfokú görebét *elliptikus görbe* nevezzük, ha a jobb oldalon álló *f(x)* függvénynek három különbözõ (komplex) gyöke van.

Tétel: Az $y^2 = f(x) = x^3 + ax^2 + bx + c$ alakban felírt harmadfokú görbe elliptikus akkor és csak akkor, ha diszkriminánsa: $-4a^3c + a^2b^2 + 18abc - 4b^3 - 27c^2 \neq 0$.

Honnan ered az elnevezés: *elliptikus görbe*? Az ellipszis ívhosszának meghatározásánál használt integrálformulában jelenik meg az $y = \sqrt{f(x)}$ képlet. Így ragadt az $y^2 = f(x)$ egyenletû görbékre *elliptikus görbe* elnevezés.

Mit is jelent másképen kifejezve, hogy a görbe elliptikus? Definíció szerint: az $y^2 = f(x)$ egyenl jobb oldalán álló *f(x)* függvénynek nincs többszörös gyöke. Írjuk az egyenletet $F(x,y) = y^2 - f(x)$ formába. Nézzük meg a parciális deriváltakat: $\frac{\partial F}{\partial x} = -f'(x)$ és $\frac{\partial F}{\partial y} = 2y$.

Definíció szerint a görbe nem szinguláris, ha nincs olyan pont a görbén, ahol mindkét parciális der eltûnik. (Ez azt jelenti, hogy minden pontban egyértelmûen létezik érintõ, azaz a görbe sima.) Tegyü hogy az (x_0, y_0) pontban eltûnik mindkét parciális derivált, azaz $y_0 = 0$ és $f(x_0) = 0$ és ezért *f(x)*-nek $f'(x)$-nek x_0 közös gyöke. Ezért x_0 kétszeres gyöke *f*-nek. Ha *f*-nek x_0 kétszeres gyöke, akkor $(x_0, 0)$ görbének szinguláris pontja.

Más szóval: az elliptikus görbék nem szinguláris, harmadfokú síkgörbék.

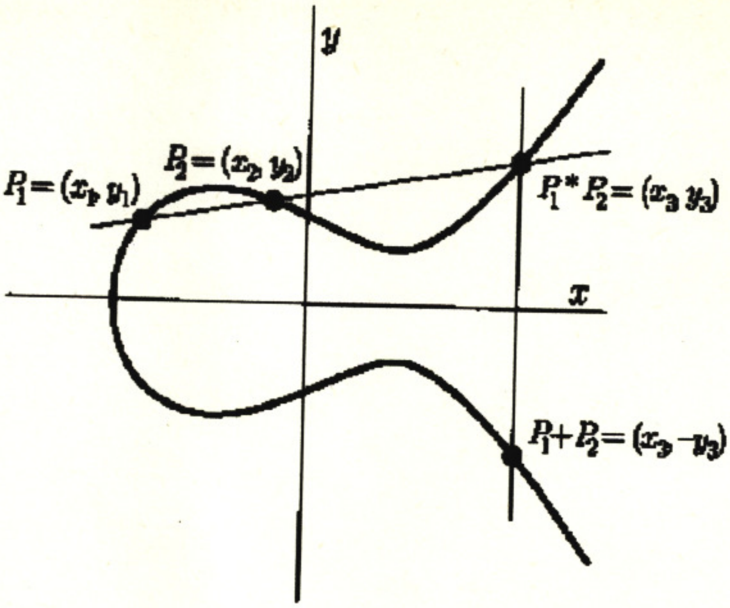
5.3. Explicit képlet az elliptikus görbék pontjainak összeadásra

Induljunk ki a Weierstrass-féle alakból, azaz $y^2 = x^3 + ax^2 + bx + c$ -bõl. Áttérve homogén koordinátákra, $x = \frac{X}{Z}, y = \frac{Y}{Z}$ helyettesítést végrehajtva és Z^3 -bel beszorozva kapjuk: $Y^2Z = X^3 + aX^2Z + bXZ^2 + cZ^3$.

$Z = 0$ -t írva az egyenletbe kapjuk, hogy $X^3 = 0$, azaz $X = 0$ háromszoros gyök. Ez azt jelenti, hog görbének egy pontja van a végtelenben (nevezetesen az *Y*-tengely végtelen távoli pontja) és ez a végtelenbeli pont inflexiós pontja a görbének. Ezért ebben a pontban vont érintõ háromszoros multiplicitással érinti a görbét. (Parciális deriváltakkal ellenõrizhetõ, hogy ez a pont nem szinguláris pont.) Tehát a Weierstrass formában lévõ görbéknek egyetlen pontja van a végtelenben és ezt nevez *O*-nak, a csoport null elemének.

Tehát görbének pontjai az *xy* affin síkon látszanak, kivéve az *O* pontot, amely az *Y*-tengely végtel távoli pontja. Most láthatjuk, hogy minden függõleges egyenes három pontban metszi a görbét; nevezetesen a végtelen távoli egyenesnek háromszoros pontja az *O*, egy függõleges egyenesnek az *x* síkon van két közös pontja a görbével plusz az *O*, a nem függõleges egyenesek pedig az *xy* síkon metszik a görbét három (nem feltétlenül különbözõ) pontban.

Ezek után elkezdhetjük közelebbrõl szemügyre venni a csoportmûveletet. Hogy is kell összeadni a Weierstrass formában lévõ görbe P_1 és P_2 pontjait (feltéve, hogy $P_1 \neq P_2$)? Elõször is meg kell húzn P_1 és P_2 pontokat összekötõ egyenest, és venni kell az egyenes és a görbe harmadik metszéspontját. megkapjuk P_1*P_2 pontot. Összekötjük P_1*P_2-t *O*-val, azaz függõleges egyenest húzunk P_1*P_2-n keresztül és vesszük a harmadik metszéspontot ami megadja a keresett P_1+P_2 pontot. Vegyük észre, hogy a Weierstrass alakban lévõ görbe *X*-tengelyre szimmetrikus, így ha P_1*P_2 koordinátái (x_3, y_3), akkor P_1+P_2 koordinátái $(x_3, -y_3)$ lesznek. Tekintsünk az ábrára:



Legyen $P_1=(x_1, y_1)$, $P_2=(x_2, y_2)$, $P_1*P_2=(x_3, y_3)$ és $P_1+P_2=(x_3, -y_3)$. Feltéve, hogy ismerjük és P_2 koordinátáit szeretnénk meghatározni (x_3, y_3) értékét.

Elõször is írjuk fel P_1, P_2 -t összekötõ egyenes egyenletét: $y=\lambda x+\nu$, ahol $\lambda=\frac{y_2-y_1}{x_2-x_1}$ és $\nu=y_1-\lambda x_1=y_2-\lambda x_2$.

Ahhoz, hogy megkapjuk a P_1, P_2 -t összekötõ egyenes görbével vett harmadik metszéspontját be helyettesítenünk az egyenes egyenletét a görbe egyenletébe.
Kapjuk: $y^2=(\lambda x+\nu)^2=x^3+ax^2+bx+c$.

Egy oldalra rendezve: $0=x^3+(a-\lambda^2)x^2+(b-2\lambda\nu)x+(c-\nu^2)$. Ennek a harmadfokú egyenletn három megoldása lesz a metszéspontok x koordinátái: x_1, x_2, x_3. Így kapjuk:

$$x^3+(a-\lambda^2)x^2+(b-2\lambda\nu)x+(c-\nu^2)=(x-x_1)(x-x_2)(x-x_3).$$

A gyökök és együtthatók közötti összefüggésbõl kapjuk: $x_3=\lambda^2-a-x_1-x_2$ és $y_3=\lambda\, x_3+\nu$.

Abban az esetben, amikor $P_1=P_2$ akkor a következõ módon változik a helyzet: a két pontot összekötõ egyenes helyett $P=(x_1, y_1)$-beli érintõt kell meghúzni; az érintõ és a görbe másik metszéspontját $Q=(x_3, y_3)$ koordinátáit kell meghatározni, majd Q-t O-val összekötõ egyenes harm metszéspontját kell venni, amely $2P=(x_3, -y_3)$.

Az algebrai számolás a következõkben módosul: $P=(x_1, y_1)$-n átmenõ érintõ iránytangensét megkapjuk ha meghatározzuk y' értékét. Az $y^2=x^3+ax^2+bx+c$ egyenletet deriválva kapjuk: $2y_1y'=3x_1^2+2ax_1+b$. Tehát a P-n átmenõ érintõ egyenlete: $y-y_1=\lambda(x-x_1)$, ahol $\lambda = \dfrac{3x_1^2 + 2ax_1 + b}{2y_1}$

Ezek után az elõzõhöz hasonló megfontolások után kapjuk: $x_3 = \lambda^2 - a - 2x_1$ és $y_3 = \lambda\,(x_3 - x_1)$.

Összefoglalásként egy formulával leírva az $y^2 = x^3 + ax^2 + bx + c$ alakban felírt elliptikus görbe pontjai között definiált '+' mûvelet: adott $P_1 = (x_1, y_1)$ és $P_2=(x_2, y_2)$ pontok az elliptikus görbén. H $= x_2$ és $y_1 = -y_2$, akkor $P_1+P_2 = O$; különben $P_1+P_2 = (x_3, y_3)$, ahol

$x_3 = \lambda^2 - a - x_1 - x_2$, $y_3 = \lambda\,(x_1 - x_3) - y_1$ és $\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{ha } P_1 \neq P_2 \\ \dfrac{3x_1^2 + 2ax_1 + b}{2y_1}, & \text{ha } P_1 = P_2. \end{cases}$

6. Elliptikus görbéken alapuló kriptorendszerek

Ebben a fejezetben megnézzük, hogy miként lehet általánosítani a negyedik fejezetben tárgyalt **diszkrét logaritmus** problémát és miként hasznosíthatjuk azt a tudásunkat, amelyet elõzõ fejezetben szereztünk az elliptikus görbék pontjai felett vett csoporttal kapcsolatban. Ennek érdekében legelõsz definiáljuk általánosan a **diszkrét logaritmus** problémát egy (véges) G csoport felett, ahol a csoportmûveletet ' ∘ '-rel jelöljük.

6.1. Az általánosított diszkrét logaritmus probléma

Legyen adott az I = (p, α, β) hármas, ahol G véges csoport a ' ∘ ' mûveletre nézve, $\alpha \in G$ és $\beta \in H$, ahol $H = \{\alpha^i$, ahol $i \geq 0$ egész $\}$, azaz H az α által generált részcsoport.
Keressük azt a $0 \leq a \leq \lvert H \rvert - 1$ egészet, amelyre $\alpha^a = \beta$, ahol α^a jelentése: $\underbrace{\alpha \circ \alpha \circ \ldots \circ \alpha}_{a\text{-szor}}$.
Ekkor a jelölje α alapú logaritmus b-t ($a = \log_\alpha b$).

Nézzük meg egy kicsit jobban az általánosított **diszkrét logaritmus** problémát! Az $\alpha \in G$ elem á generált részcsoport, H, természetesen ciklikus és a csoport rendje: $\lvert H \rvert$. Tehát a diszkrét logaritmus probléma bármely variációja bizonyos értelemben ekvivalens egy ciklikus csoportban vett diszkrét logaritmus problémával. Viszont a diszkrét logaritmus probléma bonyolultságát alapvetõen a reprezentáló csoport határozza meg.

Tekintsük például a $\mathbf{Z}_n$ additív csoportját, ahol a problémát egyszerû megoldani. Tegyük fel hogy $\alpha \in \mathbf{Z}_n$-re lnko(α, n) = 1, tehát α generáló eleme $\mathbf{Z}_n$-nek. Mivel a csoportmûvelet a modulo n összeadás, a „hatványozás" a modulo n szorzásnak fog megfelelni. Tehát ezen a csoport felett definiá diszkrét logaritmus probléma a következõképpen néz ki:

Keressük azt az a egész értéket, amelyre: $\alpha a \equiv \beta$ mod n.

Mivel lnko(α, n) = 1, ezért α-nak létezik multiplikatív inverze modulo n, tehát meg tudjuk határozni α^{-1} értékét a **kiterjesztett euklideszi algoritmussal**. Megoldásként kapjuk: $\log_{\alpha}\beta = \beta\alpha$ mod n.

Az elõbb általánosan is definiáltuk a diszkrét logaritmus problémát tetszõleges $\alpha \in G$ elem által generált részcsoportra és az elemein értelmezett ' $\circ$ ' mûveletre. Ennek a H ciklikus csoportnak a rendje $|H|$, ezért izomorf $\mathbf{Z}_{|H|}$ additív csoportjával. Az elõbb említett módon megoldható a diszkrét logaritmus probléma a $\mathbf{Z}_{|H|}$ additív csoportban. Azt érezzük, hogy megoldható a probléma (H, $\circ$)-b ha visszavezetjük a $\mathbf{Z}_{|H|}$ additív csoportban már megoldottra.

Nézzük meg, hogy miként is nézne ez ki. Az a megállapítás, hogy ($\mathbf{Z}_{|H|}$, +) izomorf (H, $\circ$)-rel azt jelenti, hogy létezik egy : ψ (H, $\circ$) $\rightarrow$ ($\mathbf{Z}_{|H|}$, +) bijekció úgy, hogy: $\psi(x \circ y) = \psi(x) + \psi(y)$ mod $|H|$.

Könnyen látható, hogy ekkor: $\psi(\alpha^a) = a\psi(\alpha)$ mod $|H|$, ahol α^a jelentése: $\underbrace{\alpha \circ \alpha \circ \ldots \circ \alpha}_{a\text{-szor}}$.

Tehát $\beta = \alpha^a \Leftrightarrow a\psi(\alpha) \equiv \psi(\beta)$ mod $|H|$. Megoldva a-ra kapjuk: $\log_{\alpha}\beta = \psi(\beta)(\psi(\alpha))^{-1}$ mod |

Következésképpen, ha találnánk egy eljárást a izomorfizmus kiszámolására, akkor lenne egy hatékony algoritmusunk a diszkrét logaritmus probléma megoldására (H, $\circ$)-ben. A helyzet viszont hogy nincs ismert és általánosan használható eljárás meghatározására. Ez a rövid tárgyalás is azt mutatja, hogy legyen a diszkrét logaritmus probléma akár bonyolult, akár egyszerû, nehézsége mindenképpen a reprezentáló csoporttól függ. Újabb csoportokat keresve ezért volt érdemes foglalk az elliptikus görbék pontjain vett csoporttal az elõzõ fejezetben.

Ezek után definiálhatjuk az általánosított **ElGamal** kriptorendszert a $H \leq G$ részcsoportban.

6.2. Az általánosított ElGamal kriptorendszer

Legyen G véges csoport a ' $\circ$ ' mûveletre nézve és legyen $\alpha \in G$ egy olyan elem, amelyre a H felett vett diszkrét logaritmus probléma megfelelõen nehéz, ahol H = {α^i, ahol $i \geq 0$ egész }, azaz H az α által generált részcsoport. Legyen továbbá $\mathcal{P} = G$, $\mathcal{C} = G \times G$, illetve $\mathcal{K}$ = {(p, α , a , β) úgy, hogy $\beta = \alpha^a$ }. Az α és a β ,értékek nyilvánosak, az a értéke pedig titkos.

Adott K = (G, α , a , β) kulcsra és egy tetszõleges, titkos $k \in \mathbf{Z}_{|H|}$ -ra legyen:

$$e_K(x, k) = (y_1, y_2),$$

ahol $y_1 = \alpha^k$ és $y_2 = x \circ \beta^k$, illetve $y = (y_1, y_2)$ kódolt üzenetre pedig:

$$d_K(y) = y_2 \circ (y_1^{a})^{-1}.$$

Megjegyzés: Elõfordulhat, hogy Alice nem ismeri a H részcsoport rendjét. Ekkor kódoláshoz válasz egy $0 \le k \le |G| - 1$ véletlen értéket is, a módszer ezen túl változtatás nélkül használható. Jegyezzük meg azt is, hogy a G csoportnak nem kell feltétlenül kommutatívnak lennie, hiszen a H részcsoportja mivel ciklikus, kommutatív lesz, mivel α hatványai felcserélhetõek.

6.3. Elliptikus görbék véges test felett

Def: Legyen $p > 3$ prím. Az $y^2 = x^3 + ax^2 + bx + c$ alakban felírt elliptikus görbe $\mathbf{Z}_p$ felett legyen azon $(x, y) \in \mathbf{Z}_p \times \mathbf{Z}_p$ pontok halmaza, amelyek kielégítik az $y^2 \equiv x^3 + ax^2 + bx + c$ mod p kongruenciát, ahol $a, b, c \in \mathbf{Z}_p$ úgy, hogy a diszkrimináns $D = -4a^3c + a^2b^2 + 18abc - 4b^3 - 27c^2$ 0 mod p, kiegészítve az Y-tengely végtelen távoli pontjával O-val.

A 7.12. Függelékben található egy Mathematica® program $\mathbf{Z}_p$ feletti E elliptikus görbe pontjainak összeadására.

Nézzünk egy példát:

Legyen az E elliptikus görbe az $y^2 = x^3 + x + 6$ a $\mathbf{Z}_{11}$ test felett. Határozzuk meg E pontjait! Tegy ezt úgy, hogy elõször $\forall x \in \mathbf{Z}_{11}$-re kiszámoljuk az $x^3 + x + 6$ mod 11 értéket, és a kapott értékek kö megnézzük melyek kvadratikus maradékok modulo 11. Az így kapott párok lesznek E pontjai $\mathbf{Z}_{11}$ fe

Jegyezzük meg azonban azt, hogy a harmadik fejezetben említett **Euler-lemma** csak arra ad válas hogy az adott x kvadratikus maradék-e vagy sem modulo p, de semmiféle segítséget nem nyújt x mo „négyzetgyökeinek" meghatározásához. Azonban ha $p \equiv 3$ mod 4 akkor létezik egy egyszerû keple kvadratikus maradék négyzetgyökének meghatározására mod p.

Tegyük fel, hogy $x \not\equiv 0$ mod p kvadratikus maradék mod p, és $p \equiv 3$ mod 4. Így azt kapjuk, hogy $(\pm x^{(p+1)/4})^2 \equiv x^{(p+1)/2} \equiv x^{(p-1)/2} x \equiv x$ mod p. Ekkor ismét alkalmazva az Euler-lemmát, mely szerint, ha x kvadratikus maradék mod p, akkor $x^{(p-1)/2} \equiv 1$ mod p. Ezért x két négyzetgyöke modu p: $\pm x^{(p+1)/4}$ mod p.

Igen érdekes, hogy $p \equiv 1$ mod 4 esetén nem ismert polinomiális idejû algoritmus egy mod p kvadratikus maradék négyzetgyökeinek meghatározására. Viszont létezik polinomiális idejû valószínûségi algoritmus.

Visszatérve E pontjainak meghatározásához $\mathbf{Z}_{11}$ felett, a számításinkat a következõ táblázatban foglalhatjuk össze:

x	$x^3 + x + 6 \bmod 11$	az eredmény kvadratikus maradék-e?	y
0	6	nem	
1	8	nem	
2	5	igen	4, 7
3	3	igen	5, 6
4	8	nem	
5	4	igen	2, 9
6	8	nem	
7	4	igen	2, 9
8	9	igen	3, 8
9	7	nem	
10	4	igen	2, 9

Így E-nek 13 pontja van. Mivel minden p prím rendû ciklikus csoport izomorf $\mathbf{Z}_p$-vel, ezért E izor $\mathbf{Z}_{13}$-mal. Tehát O-t kivéve minden elem generátor eleme E-nek. Válasszuk generátornak az α = (2, 7 pontot és határozzuk meg a hatványait.

Ahhoz, hogy meghatározzuk $2\alpha = (2, 7) + (2, 7)$ értékét elõször számoljuk ki λ-t:

$$\begin{aligned}\lambda &= (3 * 2^2 + 1)(2 * 7)^{-1} \bmod 11 \\ &= 2 * 3^{-1} \bmod 11 \\ &= 2 * 4 \bmod 11 \\ &= 8.\end{aligned}$$

Ezután meghatározzuk: $x_3 = 8^2 - 2 - 2 \bmod 11$, azaz $x_3 = 5$ és $y_3 = 8(2 - 5) - 7 \bmod 11$, azaz $y_3 =$ értékeket. Így $2\alpha = (5, 2)$.

Ezt követõen meghatározzuk $3\alpha = 2\alpha + \alpha = (5, 2) + (2, 7)$ értéket. Most is λ-t számoljuk ki elõsz

$$\begin{aligned}\lambda &= (7 - 2)(2 - 5)^{-1} \bmod 11 \\ &= 5 * 8^{-1} \bmod 11 \\ &= 5 * 7 \bmod 11 \\ &= 2.\end{aligned}$$

Majd meghatározzuk: $x_3 = 2^2 - 5 - 2 \bmod 11$, azaz $x_3 = 8$ és $y_3 = 2(5 - 8) - 2 \bmod 11$, azaz $y_3 = 3$ értékeket. Így $3\alpha = (8, 3)$.

Hasonló számításokat végezve meghatározhatjuk a fennmaradó hatványait α-nak. A következõket kapjuk:

$\alpha = (2, 7)$	$2\alpha = (5, 2)$	$3\alpha = (8, 3)$
$4\alpha = (7, 2)$	$5\alpha = (3, 6)$	$6\alpha = (7, 9)$
$7\alpha = (7, 2)$	$8\alpha = (3, 5)$	$9\alpha = (10, 9)$
$10\alpha = (8, 8)$	$11\alpha = (5, 9)$	$12\alpha = (2, 4)$.

Jelölés: Legyen E $\mathbf{Z}_p$ ($p > 3$, prím) felett definiált elliptikus görbe. Ekkor $\#E$ jelölje E pontjainak szá

$\mathbf{Z}_p$ felett.

Hasse-tétel: Legyen E $\mathbf{Z}_p$ ($p > 3$, prím) felett definiált elliptikus görbe. Ekkor

$$p+1-2\sqrt{p} \leq \#E \leq p+1+2\sqrt{p}.$$

Megjegyzés: $\#E$ értékének pontos meghatározása már egy kicsit bonyolultabb feladat, de erre is léte egy jó polinomiális idejû algoritmus: a **Schoof algoritmus**.

6.4. Példák elliptikus görbéken alapuló kriptorendszerekre

Most, hogy már meg tudjuk határozni $\#E$ értékét, ezután E-ben egy ciklikus részcsoportot szeretr találni, amelyben a diszkérét logaritmus probléma nehéz. A következõ tétel az E csoport struktúrájár mond valamit:

Tétel: Legyen E $\mathbf{Z}_p$ ($p > 3$, prím) felett definiált elliptikus görbe. Ekkor léteznek n_1, n_2 egészek úgy, hogy E izomorf $\mathbf{Z}_{n_1} \times \mathbf{Z}_{n_2}$ -nel, továbbá $n_2 \mid n_1$ és $n_2 \mid (p - 1)$.

Megjegyzés: Ha $n_2 = 1$, illetve $\#E$ prím, vagy különbözõ prímek szorzata, akkor E ciklikus csoport.

Tehát, ha az n_1, n_2 egészek értékét meg tudjuk határozni, akkor azt is tudjuk, hogy E-nek létezik $\mathbf{Z}$ gyel izomorf részcsoportja, amelyet használhatnánk az általánosított **ElGamal** kriptorendszerhez.

6.4.1. Az ElGamal módszer elliptikus görbék felett

Nézzünk egy példát az **ElGamal** rendszerre, amely az elõzõ példában tárgyalt elliptikus görbét használja:

Tegyük fel, hogy $\alpha = (2, 7)$ és Bob titkos „kitevõje" a 7, tehát $b = 7\alpha = (7, 2)$. Így a kódoló függvén következõ lesz:

$$e_K(x, k) = (k(2, 7), x + k(2, 7)),$$

ahol $x \in E$ és $0 \leq k \leq 12$ egész, illetve a dekódoló függvény:

$$d_K(y_1, y_2) = y_2 - 7y_1.$$

Tegyük fel, hogy Alice az $x = (10, 9)$ üzenetet akarja titkosítani (amely E egy pontja). Ha Alice által választott titkos kitevõ a $k = 3$, akkor meghatározza az $y_1 = 3(2, 7) = (8, 3)$ és az $y_2 = (10, 9) + (3,$ $(10, 2)$ értékeket és megkapja a kódolt üzenetet az $y = ((8, 3),(10, 2))$ -t, amit elküld Bobnak.

Miután Bob megkapta az y kódolt üzenetet a következõ számítást végzi el: $x = (10, 2) - 7(8, 3) = (1$ $2) + (6, 3) = (10, 9)$, így visszakapja az eredeti üzenetet.

6.4.2. Az elliptikus ElGamal hibája

Az elsõ dolog az, hogy a nyílt üzenetek halmaza az E görbe pontjai és nem ismert determinisztiku algoritmus E pontjainak meghatározására. Még komolyabb probléma, hogy míg a $\mathbf{Z}_p$ feletti **ElGama** algoritmusnak kétszeres expanziós tényezõje van; azaz a kódolt üzenet az eredeti üzenet hosszának kétszerese, az elliptikus görbék feletti implementáláskor viszont kb. négyszeres lesz az expanziós tényezõ. Ez azt jelenti, hogy az implementálás lassul és a kódolt üzenet hossza szükségtelenül megnő

Az **ElGamal** módszernek egy a gyakorlatban jobban használható változata a **Menezes-Vanstone** módszer.

6.4.3. A Menezes-Vanstone módszer elliptikus görbék felett

A **Menezes-Vanstone** módszer formális leírása:

Legyen E $\mathbf{Z}_p$ ($p > 3$, prím) felett definiált elliptikus görbe úgy, hogy H az E-nek egy olyan ciklikus részcsoportja, amelyben a diszkrét logaritmus probléma nehéz. Legyen továbbá $\mathcal{P} = \mathbf{Z}_p^* \times \mathbf{Z}_p^*$, $\mathcal{C} = E \times \mathbf{Z}_p^* \times \mathbf{Z}_p^*$, illetve $\mathcal{K} = \{(E, \alpha, a, \beta)$ úgy, hogy $\beta = a\alpha\}$. Az α és a β, értékek nyilvánosak, az a értéke pedig titkos. Adott $K = (E, \alpha, a, \beta)$ kulcsra, egy tetszõleges, titkos $k \in \mathbf{Z}_{|H|}$ -ra és $x = (x_1, x_2) \in \mathbf{Z}_p^* \times \mathbf{Z}_p^*$ -ra legyen:

$$e_K(x, k) = (y_0, y_1, y_2),$$

ahol $y_0 = k\alpha$, $(c_1, c_2) = k\beta$, $y_1 = c_1x_1 \bmod p$ és $c_2 = c_2x_2 \bmod p$.

Az $y = (y_0, y_1, y_2)$ kódolt üzenetre pedig:

$$d_K(y) = (y_1c_1^{-1} \bmod p; y_2c_2^{-1} \bmod p),$$

ahol $ay_0 = (c_1, c_2)$.

A következõ példa erejéig térjünk vissza az $y^2 = x^3 + x + 6$ görbéhez $\mathbf{Z}_{11}$ felett. Azt látjuk, hogy a **Menezes-Vanstone** kriptorendszer 10*10 = 100 különbözõ nyílt üzenetet kódolhat, az eredeti móds 13 üzenetével szemben.

Tegyük fel, ahogy az elõbbi példában is tettük, hogy $\alpha = (2, 7)$ és Bob titkos „kitevõje" a 7, tehát $7\alpha = (7, 2)$. Tegyük fel továbbá, hogy Alice az $x = (x_1, x_2) = (9, 1)$ üzenetet szeretné kódolni, amely E egy pontja és a véletlenül választott érték: $k = 6$.

Alice elõször meghatározza az $y_0 = k\alpha = 6(2, 7) = (7, 9)$ értéket, majd a $k\beta = 6(7, 2) = (8, 3)$ érté így adódik: $c_1 = 8$ és $c_2 = 3$. Ezt követõen kiszámolja az $y_1 = c_1x_1 \bmod p = 8*9 \bmod 11 = 6$ és az y $c_2x_2 \bmod p = 3*1 \bmod 11 = 3$ értékeket. Az így kapott $y = (y_0, y_1, y_2) = ((7, 9), 6, 3)$ kódolt üzene elküldi Bobnak.

Miután Bob megkapta az y üzenetet elõször kiszámolja a $(c_1, c_2) = ay_0 = 7(7, 9) = (8, 3)$ értéket, m az $x = (y_1c_1^{-1} \bmod p ; y_2c_2^{-1} \bmod p) = (6*8^{-1} \bmod 11; 3*3^{-1} \bmod 11) = (6*7 \bmod 11; 3*4 \bmod 11)$ (9, 1)értéket. Ezzel helyesen visszakapja az eredeti üzenetet.

6.5. Az elliptikus görbék biztonságáról

Ebben a részben az elliptikus görbéken alapuló kriptorenszerek biztonságának elvi hátteréről lesz : illetve biztonságát összehasonlítjuk más rendszerek biztonságával. Ezt követõen a gyakorlati felhasználásra adunk javaslatot.

6.5.1. Biztonság elvi lehetõségei

A negyedik fejezetben említettünk néhány, a diszkrét logaritmus probléma megoldására készült algoritmust. A **Shanks** és a **Pohlig-Hellman** algoritmusuknak létezik az elliptikus görbék csoportjár vonatkozó adaptációja, de a leghatékonyabbnak tartott **Index Kalkulus** eljárásnak eddig még nem ismert elliptikus görbéken vett megfelelõje. Azonban létezik egy eljárás, amely kihasznál egy explicit izomorfizmust az elliptikus görbék és a véges testek között, és hatékony algoritmushoz vezet az elliptikus görbék egy bizonyos osztályánál. A technika **Menezes**, **Okamoto** és **Vanstone** nevéhez fûzõdik és különösen a szuperszinguláris elliptikus görbék körében használható. Viszont, ezek a görl könnyen elkerülhetõek. A mai lehetõségek szintjén biztonságosnak tûnnek az olyan algebrai szempontból általános elliptikus görbék, melyek rendjének van nagy prímtényezõje. Ezekre már nem hatékony a **Pohlig-Hellman** algoritmus.

6.5.2. Összehasonlítás más rendszerekkel

A kriptográfiai algoritmusok egy része bizonyos matematikai kérdések bonyolultságán alapszik. A kriptográfusok fordított szemlélettel nézik a bonyolultsági eredményeket, elsõsorban a hatékonyan n kezelhetõ problémák érdekesek számukra. Nézzük meg, hogy melyek azok a problémák, amelyekre : dolgozatomban szereplõ kriptorendszerek épülnek:

1. egész számok faktorizációjának problémája (**IFP**)
2. diszkrét logaritmus probléma (**DLP**)
3. elliptikus görbéken alapuló diszkrét logaritmus probléma (**ECDLP**)

A kriptorendszerek feltörésének két módja: software és célhardware útján. Elõször nézzük meg a software nyújtotta lehetõségeket.

Software adta lehetõségek:

Tegyük fel, hogy 1 **MIPS** számítógép 4×10^4 elliptikus összeadást tud végezni másodpercenként.] az egy év alatt végrehajtható elliptikus összeadások száma (**MIPS year**):

$$(4\times10^4)*(60\times60\times24\times365) \approx 2^{40}.$$

A következõ táblázat azt mutatja, hogy a **Pollard rho**-eljárással mennyi idõ szükséges különbözõ ekre az elliptikus logaritmus értékének meghatározására (forrás: The ECC Tutorials and Whitepaper [2]):

Test mérete(bitben)	n mérete(bitben)	$\sqrt{\pi n/2}$	MIPS year
163	160	2^{80}	$9,6\times10^{11}$
191	186	2^{93}	$7,9\times10^{15}$
239	234	2^{117}	$1,6\times10^{23}$
359	354	2^{177}	$1,5\times10^{41}$

431	426	2^{213}	$1,0\times10^{52}$

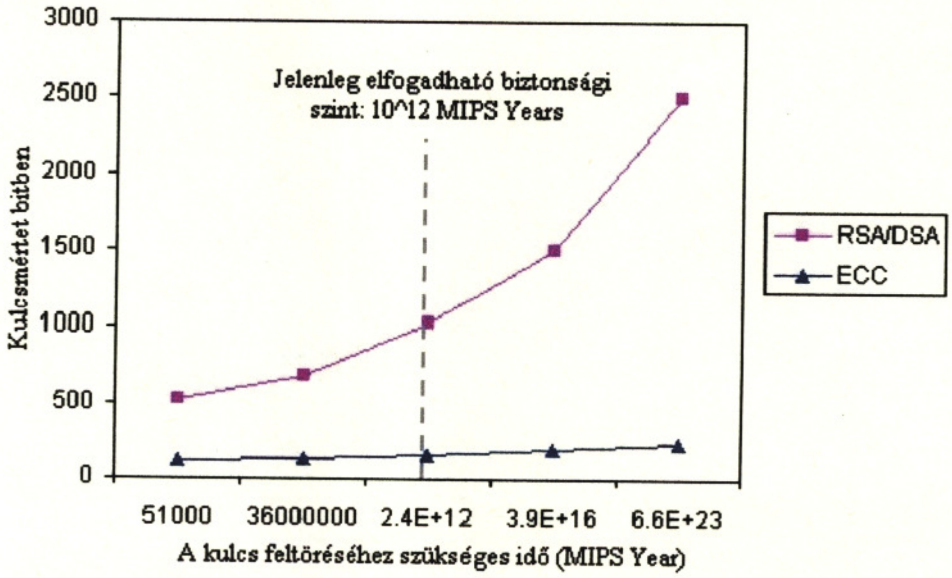
Például, ha 10.000 számítógépek kötünk össze, amelyek egyenként 1.000 **MIPS** teljesítményre képesek, akkor egy $n \approx 2^{160}$ nagyságrendû szám esetén az elliptikus logaritmus 96.000 év alatt határozható meg.

Összehasonlításként a következõ táblázat azt mutatja, hogy körülbelül mennyi idõ szükséges a ma algoritmusokkal egy n szám faktorizációjához, amely durván ekvivalens a diszkrét logaritmus meghatározásával modulo egy 1024 bites p prím (forrás: The ECC Tutorials and Whitepapers [2]).

Felbontandó egész mérete(bitben)	MIPS year
512	3×10^{4}
768	2×10^{8}
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

Végül egy diagrammal szemléltetve a fent említett algoritmusok közötti különbséget (forrás: The ECC Tutorials and Whitepapers [2]):

Kriptorendszerek biztonsági összehasonlítása
Elliptikus görbék biztonsága (ECC) kontra DES/DSA



Jelenlegi ismereteink az **IFP**, **DLP** és **ECDLP** problémákra ismert algoritmusokról azt a kijelentést támasztják alá, hogy az **ECDLP** probléma lényegesen nehezebb mind az **IFP** mind a **DLP** problémáknál.

Hardware adta lehetõségek:

Sokkal ígéretesebbnek látszik a támadás hardware-es útja. Egy speciálisan erre a célra épített hardware, amely párhuzamosan futtatja a **Pollard rho**-eljárást. Feltételezések szerint, ha n $\approx 10^{36}$ és célhardware $m = 325.000$ processzorral dolgozik párhuzamosan, amely gép építési költsége nagyjáb 10 millió dollár, a diszkrét logaritmust képes lenne körülbelül 35 nap alatt meghatározni.

6.5.3. Javaslat gyakorlati felhasználásra

1996. januárjában Matt Blaze, Whitfield Diffie, Ronald L. Rivest, Bruce Schneier, Tsutomu Shimom Eric Thompson és Michael Wiener közös publikációjukban [1] javaslatot tettek szimmetrikus kulcsú rendszerekben használt kulcsok minimális méretére (ilyen például a DES):

> *„Ahhoz, hogy megfelelõ védelmet biztosítson a legkomolyabb veszélyekkel - tõkeerõs kereskedelmi vállalatok vagy kormányzati hírszerzõ hivatallal szemben - adataink védelme érdekében jelenleg legalább 75 bites kulcs szükséges. Ahhoz, hogy az elkövetkezendõ 20 éven belül, figyelembe véve a technika fejlõdését, az újonnan fejlesztett rendszereknek legalább 90 bites kulcshosszal kell rendelkezniük."*

Ezt a megállapítást alkalmazva az elliptikus görbéken alapuló rendszerekre azt látjuk, hogy rövid távon legalább 150 bites kulcs, középtávon legalább 180 bites kulcs használata ajánlott.

7. Függelék - algoritmusok és ábrák gyûjteménye

7.1. A kiterjesztett euklideszi algoritmus pszeudokódja

$n_0 = n$ {a modulus}

$b_0 = b$ {az elem, amelynek a multiplikatív inverzét keressük mod n}

$t_0 = 0$

$t = 1$

$q = \left\lfloor \frac{n_0}{b_0} \right\rfloor$

$r = n_0 - q \times b_0$

ciklus amíg $r > 0$

$temp = t_0 - q \times t$

feltétel ha $temp \geq 0$ **akkor** $temp = temp \bmod n$ **feltétel vége**

feltétel ha $temp < 0$ **akkor** $temp = n - ((-temp) \bmod n)$ **feltétel vége**

$t_0 = t$

$t = temp$

$n_0 = b_0$

$b_0 = r$

$$q = \left\lfloor \frac{n_0}{b_0} \right\rfloor$$

$r = n_0 - q \times b_0$

ciklus vége

feltétel ha $b_0 \neq 1$ **akkor** *b*-nek nincs multiplikatív inverze modulo *n*

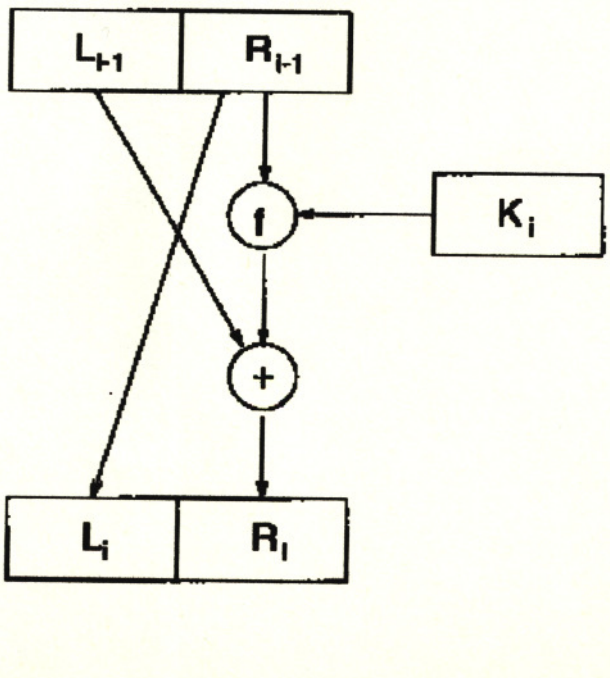
különben $b^{-1} = t$ mod *n*

feltétel vége

7.2. Az ismételt négyzetre emelés algoritmusának pszeudokódja

x = *alap*
b = *kitevõ*
n = *modulus*
$l = 1$ {a kitevõ 2-es számrendszerbeli jegyeinek száma}
ciklus amíg $b > 0$
$l = l + 1$
$b_l = b$ mod 2
$b = b$ div 2
ciklus vége
$z = 1$
ciklus amíg $l > 0$
$z = z^2$ mod n
feltétel ha $b_l = 1$ **akkor** $z = z \times x$ mod *n* **feltétel vége**
ciklus vége
ki *z* {z változó tartalmazza xb mod n értékét}

7.3. A DES egy lépésének az ábrája



7.4. A DES *IP* illetve az IP^{-1} bitpermutáló mátrixa

IP

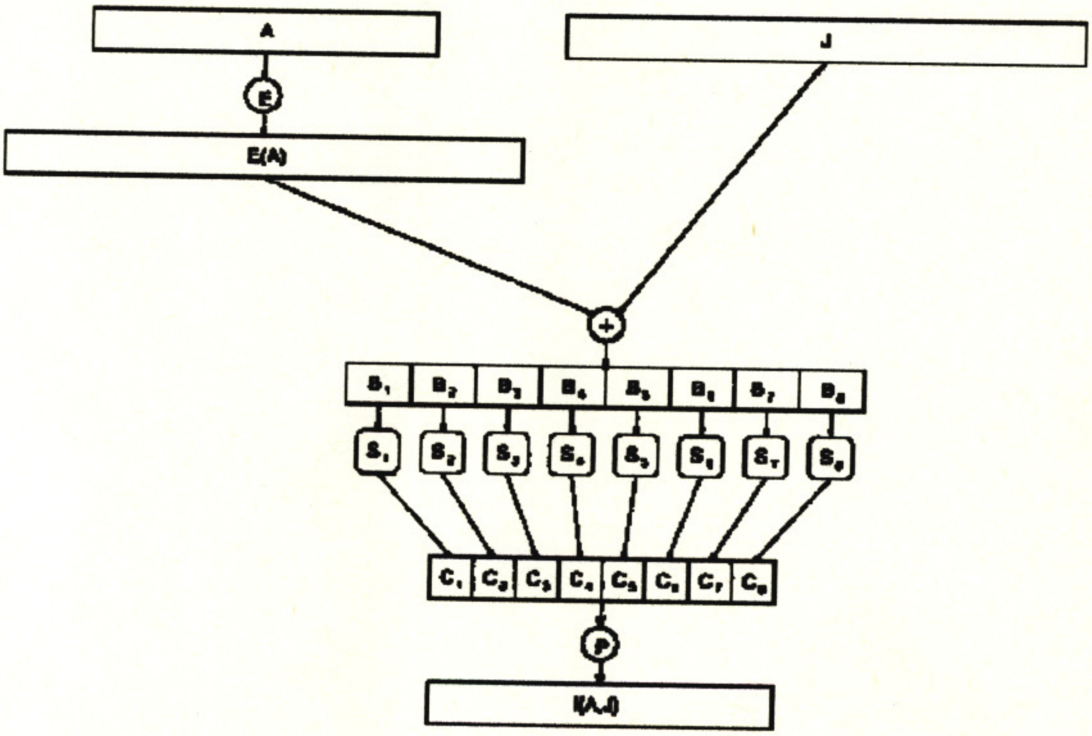
58	50	52	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}

40	8	48	16	56	24	64
39	7	47	15	55	13	63
38	6	46	14	54	22	62
37	5	45	13	53	21	61
36	4	44	12	52	20	60
35	3	43	11	51	19	59
34	2	42	10	50	18	58
33	1	41	9	49	17	57

A táblázatban szereplõ értékek az *x* nyílt szöveg adott helyen álló bitjét jelentik.

7.5. A DES f - függvénye



7.6. A DES *E* kiterjesztési függvénye

E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

7.7. A DES S mátrixai

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

7.8. A DES *P* permutációmátrixa

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

7.9. A DES *PC-1* és *PC-2* permutáció mátrixai

PC-1

57	49	41	33	25	17	9
1	58	50	41	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	10	12	4

PC -2

14	7	11	24	1
3	28	15	6	21
23	19	12	4	26
16	7	27	20	13
41	52	31	37	47
30	40	51	45	33
44	49	39	56	34
46	42	50	36	29

7.10. A DES 16 kulcstáblázata

K_1

10	51	34	60	49	17	33	57	2	9	19	42
3	35	26	25	44	58	59	1	36	27	18	41
22	28	39	54	37	4	47	30	5	53	23	29
61	21	38	63	15	20	45	14	13	62	55	31

K_2

2	43	26	52	41	9	25	49	59	1	11	34
60	27	18	17	34	50	51	58	57	19	10	33
14	20	31	46	29	63	39	22	28	45	15	21
53	13	30	55	7	12	37	6	5	54	47	23

K_3

51	27	10	36	25	58	9	33	43	50	60	18
44	11	2	1	49	34	35	42	41	3	59	17
61	4	15	30	13	47	23	6	12	29	62	5
37	28	14	39	54	63	21	53	20	38	31	7

K_4

35	11	59	49	9	42	58	17	27	34	44	2
57	60	51	50	33	18	19	26	25	52	43	1
45	55	62	14	28	31	7	53	63	13	46	20
21	12	61	23	38	47	5	37	4	22	15	54

K_5

19	60	43	33	58	26	42	1	11	18	57	51
41	44	35	34	17	2	3	10	9	36	27	50
29	39	46	61	12	15	54	37	47	28	30	4
5	63	45	7	22	31	20	21	55	6	62	38

K_6

3	44	27	17	42	10	26	50	60	2	41	35
25	57	19	18	1	51	52	59	58	49	11	34
13	23	30	45	63	62	38	21	31	12	14	55
20	47	29	54	6	15	4	5	39	53	46	22

K_7

52	57	11	1	26	59	10	34	44	51	25	19
9	41	3	2	50	35	36	43	42	33	60	18
28	7	14	29	47	46	22	5	15	63	61	39
4	31	13	38	53	62	55	20	23	37	60	6

K_8

36	41	60	50	10	43	59	18	57	35	9	3
58	25	52	51	34	19	49	27	26	19	44	2
12	54	61	13	31	30	6	20	62	47	45	23
55	15	28	22	37	46	39	4	7	21	14	53

K_9

57	33	52	42	2	35	51	10	49	27	1	60
50	17	44	43	26	11	41	19	18	9	36	59
4	46	53	5	23	22	61	12	54	39	37	15
47	7	20	14	29	38	31	63	62	13	6	45

K_{10}

41	17	36	26	51	19	35	59	33	11	50	44
34	1	59	28	10	60	25	3	2	58	49	43
55	30	37	20	7	6	45	63	38	23	21	62
31	54	4	61	13	22	15	47	46	28	53	29

K_{11}

25	1	49	10	35	3	19	43	17	60	34	57
18	50	41	11	59	44	9	52	51	42	33	27
39	14	21	4	54	53	29	47	22	7	5	46
15	38	55	45	28	6	62	31	30	12	37	13

K_{12}

9	50	33	59	19	52	3	27	1	44	18	41
2	34	25	60	43	57	58	36	35	26	17	11
23	61	5	55	38	37	13	31	6	54	20	30
62	22	39	29	12	53	46	15	14	63	21	28

K_{13}

58	34	17	43	3	36	52	11	50	57	2	25
51	18	9	44	27	41	42	49	19	10	1	60
7	45	20	39	22	21	28	15	53	38	4	14
46	6	23	13	63	37	30	62	61	47	5	12

K_{14}

42	18	1	27	52	49	36	60	34	41	51	9
35	2	58	57	11	25	26	33	3	59	50	44
54	29	4	23	6	5	12	62	37	22	55	61
30	56	7	28	47	21	14	46	45	31	20	63

K_{15}

26	2	50	11	36	33	49	44	18	25	35	58
19	51	42	41	60	9	10	17	52	43	34	57
38	13	55	7	53	20	63	46	21	6	39	45
14	37	54	12	31	5	61	30	29	15	4	47

K_{16}

18	59	42	3	57	25	41	36	10	17	27	50
11	43	34	33	52	1	2	9	44	35	26	49
30	5	47	62	45	12	55	38	13	61	31	37
6	29	46	4	23	28	53	22	21	7	63	39

7.11. A Miller-Rabin prímteszt Mathematica® kódja

eljárás: bc {n-et alakítja át $2^b c$ alakba}
bemeneti paraméter: n_ - az átalakítandó szám
eredmény: b a kettes kitevõje, c a páratlan szorzótényezõ

```
bc[n_] := Module[ {b = 0, c = n},
    While[ Mod[c, 2] ==  0, b++; c = Quotient[c, 2] ];
    {b, c} ]
```

eljárás: cnp {a tényleges teszt}
bemeneti praméterek: n_ - a tesztelni kívánt szám,
a_ - 2 és (n - 2) közé esõ véletlen érték, amire a tesztet futtatjuk,
b_ - a bc[n_] eljárás által számított kitevõ,
c_ - a bc[n_] eljárás által számított páratlan szorzótényezõ,
eredmény: „True", ha összetettnek találja; „False", ha prímnek

```
cnp[n_, a_, b_, c_] := Module[ {ex = c, pr = 1, po = a, twos = b},
   While[ ex > 0,
        If [ Mod[ex, 2] == 1, pr = Mod[pr * po, n], ];
        po = Mod[po * po, n];
        ex = Quotient[ex, 2];
        If[ po == n - 1, Return[False], ];
        If[ po == 1, Return[True], ];
   ];
   If[ pr == n - 1, Return[False], ];
   If[ pr == 1, Return[False], ];
   While[ twos > 1,
        pr = Mod[pr*pr, n];
        twos--;
        If[ pr == n - 1, Return[False], ];
        If[ pr == n - 1, Return[True], ];
   ];
   If[ (pr == 1) | | pr == n - 1, Return[False],  Return[True]]; ]
```

fõeljárás - ProbablyPrime
bemeneti paraméterek: n_ - a tesztelni kívánt szám,
ntrials_ - ismétlések száma
eredmény: „True", ha prímnek találja; „False" ha összetettnek

```
ProbablyPrime[n_, ntrials_] := Module[ {bcres = bc[n -1]},
  For[i = 0, i < ntrials, i++,
     If[cnp[n, Random[Integer,{2, n - 2}], First[bcres], Last[bcres]], Return
[False],]];
     Return[True]
  ]
```

7.12. Összeadás a $\mathbf{Z}_p$ ($p > 3$ prím) felett $y^2 = x^3 + ax^2 + bx + c$ alakban felírt

elliptikus görbe pontjain Mathematica® kód

főeljárás - EllipticAdd
bemeneti praméterek: p_ - a modulus,
a_ - a görbe egyenletében az x^2 együtthatója,
b_ - a görbe egyenletében az x együtthatója,
c_ - a görbe egyenletében a konstans tag,
P_List - az egyik összeadandó koordinátái $P = \{x_1, y_1\}$,
Q_List - a másik összeadandó koordinátái $Q = \{x_2, y_2\}$,
eredmény: az összeg pont koordinátái $R = \{x_3, y_3\}$

```
EllipticAdd[p_, a_, b_, c_, P_List, Q_List] := Module[ {lam, x3, y3, P3},
    If[ P=={O}, R = Q,
    If[ Q=={O}, R = P,
    If[ P[[1]] != Q[[1]],
        { lam = Mod[(Q[[2]]-P[[2]] * PowerMod[Q[[1]]-P[[1]], p -2, p], p];
          x3 = Mod[lam2 - a -P[[1]] - Q[[1]], p];
          y3 = Mod[- (lam(x3 - P[[1]]) + P[[2]], p];
          R = {x3, y3}
        };
    If[ (P==Q) ^ (P != {O});
        { lam = Mod[(3 * P[[1]]2 * 2 * a * P[[1]] + b) * PowerMod[2 * P[[2]], p
-2, p], p];
          x3 = Mod[lam2 - a -P[[1]] - Q[[1]], p];
          y3 = Mod[- (lam(x3 - P[[1]]) + P[[2]], p];
          R = {x3, y3}
        };
    If[ (P[[1]] == Q[[1]]) ^ (P[[2]] != Q[[2]]), R = {O}]]]]];
R]
```

7.13. Részletes adatok az RSA-155 faktorizciójáról

1999. augusztus 22-én jelentették be az amszterdami CWI épületében, egy sajtóértekezlet keretéb hogy egy faktorizációs versenyre kiírt 155 decimális jegyû (512 bites) számot sikerült prímtényezõkr bontani. A hír fontosságát jelzi, hogy komoly kutatók, számítástechnikusok, a bankok és kormányza hivatalok képviselõi is részt vettek a sajtóértekezleten. Ugyanis az eredmény szorosan összekapcsolć az RSA biztonságosságával és a jelen faktorizációs technikájának csúcsteljesítményérõl van szó. A sz a következõ volt:

RSA-155 =
109417386415705274218097073220403576120037329454492059909138421314763499842889
\
717997257891267332497625752899781833797076537244027146743531593354333

A két prímtényezõje, amelyek egyenként 78 jegyûek:

p_1 = 102639592829741105772054196573991675900716567808038066803341933521790711307

$p_2 =$ 1066034883801684548209272203600128786792079585759892915222706082371930628 08

Az eljárásban használt algoritmus a számtest szita, angolul a Number Field Sieve (NFS) volt. A felhasznált hardware:

160 db	175-400 MHz SGI és Sun workstation
8 db	250 MHz SGI Origin 2000 processzor
120 db	300-450 MHz Pentium II PC
4 db	500 MHz Digital/Compaq boxes

és az Amsterdam Academic Computing Centre SARA Cray C916-os szuperszámítógépe.

Összességében mintegy 8000 MIPS year kapacitás.

8. Irodalomjegyzék

[1] Matt Blaze - Whitfield Diffie - Ronald L. Rivest - Bruce Schneier - Tsutomu Shimomura - Eric Thompson és Michael Wiener

Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security, 1996 jai

Elérhetõ a http://theory.lcs.mit.edu/~rivest/publications.html címen.

[2] Certicom Corp. Cryptographic Technologies

The ECC Tutorials and Whitepapers

Elérhetõ a http://www.certicom.com/research/white.htmlcímen.

[3] David Eisenbud - Mark Green és Joe Harris, Bulletin of the AMS 33. száma, 1996. július (295-3: oldal)

Cayley-Bacharach Theorems and Conjectures

Elérhetõ a http://www.ams.org/bull/1996-33-03/S0273-0979-96-00666-0/S0273-0979-96-0066€ 0.pdf címen.

[4] Fuchs László, Tankönykiadó 1977.

Bevezetés az algebrába és számelméletbe

[5] Gyarmati Edit - Turán Pál, Nemzeti Tankönyvkiadó Budapest 1994.

Számelmélet

[6] Ivanyos Gábor - Rónyai Lajos - Szabó Réka, Typotex 1998.

Algoritmusok ISBN: 963-9132-16-0

[7] Ueli M. Mauer - Stefan Wolf, Institute for Theoretical Computer Science

On the Complexity of Breaking the Diffie-Hellman Protocol

Elérhetõ a http://www.inf.ethz.ch/department/TI/um/personal/publications-maurer.htmlcímen.

[8] Alfred J. Menezes - Paul C. an Oorschot - Scott A. Vanstone, CRC Press 1996.

Handbook of Applied Cryptography ISBN: 0-8493-8523-7

[9] Henk Nieland, ERCIM News 39. száma, 1999. október (38. oldal)

Security of E-commerce by 512-bit Number Factorization

Elérhetõ a http://www.ercim.org/publication/Ercim_News/enw39/512.html címen.

[10] RSA Security Inc. - Factorization of RSA-155

Elérhetõ a http://www.rsasecurity.com/rsalabs/challenges/factoring/rsa155.html címen.

[11] Joseph H. Silverman - John Tate, Springer-Verlag, New York Inc. 1992.

Rational Points on Elliptic Curves ISBN: 3-540-97825-9

[12] Douglas R. Stinson, CRC Press 1995.

Cryptography - Theory and Practice ISBN: 0-8493-8521-0

[13] Henk von Tilborg, Eindhoven University of Technology (kézirat)

An Interactive Introduction to Cryptography

8.1. Egyéb elektronikus források (internet címek)

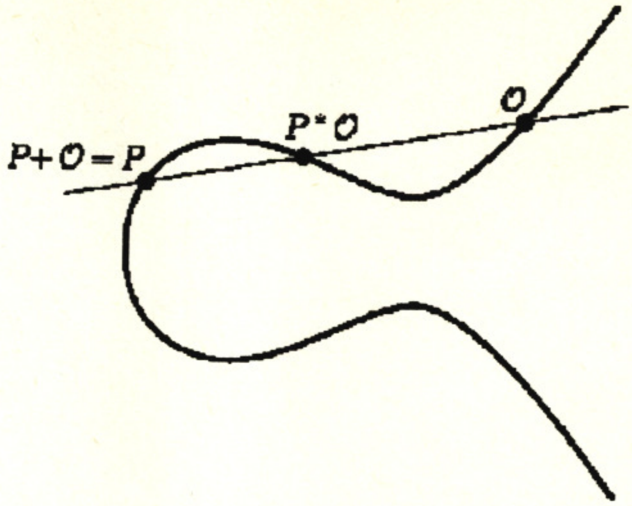
[14] Bulletin of the American Mathematical Society, News Series
http://www.ams.org/journals/bull/

[15] Certicom Corp. Cryptographic Technologies
http://www.certicom.com/

[16] ERCIM - the European Research Consortium for Informatics and Mathematics - News
http://www.ercim.org/publication/Ercim_News/backlist.html

[17] Mathematica® by Wolfram Research Inc.
http://www.wri.com/

[18] RSA Security Inc.
http://www.rsasecurity.com/


P+O=P
P*O
O